제1권 통권1호 1988년12월1일 발행(매월 1일 발행) 1988년10월10일 등록(등록번호 라-3794) ㊝140-150 서울특별시 용산구 갈월동 17

# 하이테크 퍼스널컴퓨터

8812

창•간•호

● 워드프로세서 세계의 신기원을 이룩한 **젬워드**

● 잘 알려져 있지 않은 매킨토시의 내부구조를 파헤친다.

## 매킨토시의 아키텍처(1)

유틸리티/ **노턴 유틸리티 V4.0**

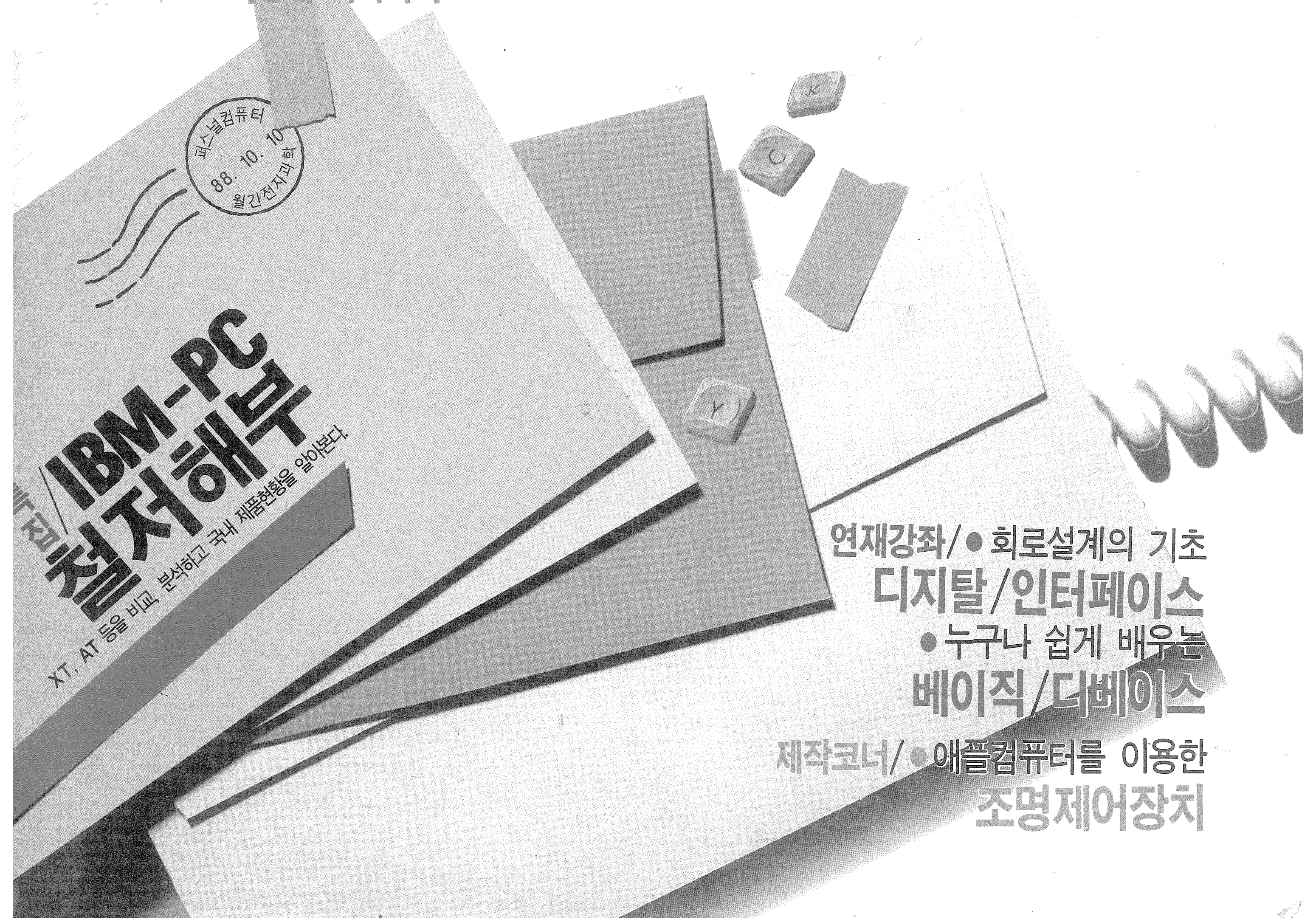
취재/ **제6회 마이크로로보트 경연대회**

구입가이드/ PC에 관한 모든 것을 총망라한 만물상

**제품의 종류/가격/구입요령/간단한 응급처치요령/구동방법**

초보자 페이지/ 누구나 손쉽게 컴퓨터를 배울 수 있다.

**어떻게 시작하나요**




연재강좌/● 회로설계의 기초

**디지탈/인터페이스**

● 누구나 쉽게 배우는

**베이직/디베이스**

제작코너/● 애플컴퓨터를 이용한

**조명제어장치**

□ 창간사 □



# 정보화사회를 앞당기는 버팀목이 되고자

오늘날의 사회를 일컬어 이른바 정보화사회라고 합니다.
어디를 가든지 화제속에 컴퓨터에 대한 용어들이 한 두가지씩은 섞여 나오며,
신문이나 잡지 등에서는 거의 매일같이 이에 관한 기사를 다루고 있읍니다.
그만큼 우리 생활 주변에서 컴퓨터를 많이 접하고 있다고 할 수 있읍니다.
그러나 그렇게 매일같이 피부로 접하면서도 컴퓨터에 대해
제대로 이해하고 있는 사람이 거의 없다는 사실은 매우 놀라운 일입니다.
2000년대에 컴퓨터를 1천만대나 보급할 계획을 세우고 있는 정부에서도
이에 대한 뾰쪽한 방안이 없다는 사실도 또한 매우 안타까운 일입니다.
퍼스널컴퓨터가 국내에 소개되고 보급되기 시작한 것은 불과 7~8년전의 일로 국내 퍼스널컴퓨터의
역사는 이제 걸음마 단계를 지나 국민학교에 입학할 정도의 나이가 되었다고 할 수 있읍니다.
따라서 이제부터는 체계적으로 이들에게 컴퓨터에 대해서 가르쳐 줘야 할 때라고 생각됩니다.
저희 잡지는 컴퓨터를 알고자 하는 모든 사람들에게 참된 지식을 심어주어 올바르게 컴퓨터를
이해하고 배울 수 있도록 모든 노력을 다하고자 합니다.
많은 사람들이 컴퓨터를 제대로 활용한다면 저희의 보람은 그만큼 더욱 더 커질 것이고,
또한 정보화사회를 앞당기는 지름길이 될 것입니다.
아무쪼록 최선을 다해 노력하고자 하오니 많은 지도편달 있으시길 바랍니다.
감사합니다.

1988년 11월 23일

월간 퍼스널컴퓨터

발행인 곽 종 한

# 어떻게 시작하나요

컴퓨터의 역사는 30여년에 불과하지만 오늘날 컴퓨터는 우리 생활 전영역에 걸쳐 영향을 미치고 있어 컴퓨터가 없는 생활은 상상하기 조차 어렵게 되었다.

이런 폭발적인 인기와 더불어 사용이 증가되고 있음에도 불구하고 사람들은 대부분 '전자두뇌'라는 관념적인 단어로만 컴퓨터를 인식하고 있을 뿐이다. 이것은 사람들이 컴퓨터에 대한 관심과는 반비례로 그 활용방법을 잘알지 못하기 때문이라고 생각된다.

이런 이유로 컴퓨터(여기에서는 IBM PC)를 처음 접하는 사람들을 위해 컴퓨터를 이해하는데 필요한 사항과 현상들을 알기 쉽게 설명하고자 한다.

## 시작은

PC-DOS는 IBM PC의 자원(시스템 유니트, 디스크, 인터페이스 카드 등) 운용을 보조하기 위한 일종의 프로그램이라고 할 수 있다.

IBM의 PC-DOS가 출현하기 까지의 경위를 간단히 소개하면, 퍼스널컴퓨터용 OS의 지최는 디지탈리서치사가 개발한 CP/M이었다. 이것은 후에 표준 OS로서 자리잡아 거의 대부분의 기종이 CP/M을 표준 DOS로 채택하였다. MS-DOS도 CP/M을 모델로 하여 마이크로소프트사에 의해서 개발되었는데 IBM PC에서 표준 DOS로 채택하면서 PC-DOS라는 이름으로 널리 보급됐다.

이러한 PC-DOS를 제대로 사용하기 위해서는 우선 디스켓을 어떻게 다루어야 하며 어떤 성질을 가지고 있는지를 잘 알아야 한다.

## 디스켓 사용시 유의사항

### ●보관방법

디스켓을 종이봉투에서 빼낸다음 디스크드라이브 도어를 제거한다. 레이블이 위를 향하게 한 후 디스켓을 드라이브에 넣는다. 디스켓이 끝까지 들어갔는지 확인후 드라이브 도어를 닫는다. 넣을 때 잘 들어가지 않으면 디스켓을 빼내고 다시 넣어 본다. 이때 강제로 넣게 되면 디스켓이나 디스크드라이브가 손상되므로 조심해야 한다(그림 1).

사용중 실수로 디스켓을 지우거나 손상을 입힐 경우 중요한 데이터가 상실될 가능성이 있으므로 반드시 이럴 경우는 백업본을 만들어 놓아야 한다. 그리고 디스켓 사용시 실수로 같은 화일명으로 세이브시킬 경우 이미 들어있던 데이터가 분실되므로 이럴 경우를 대비해서 라이트 방치를 위해서 노치를 실버탭을 이용해서 막는다.

디스켓은 디스켓용 봉투에 보관하고 사용하지 않을 때는 디스켓에 딸린 박스에 보관한다. 이렇게 하여야만 자성체로부터 디스켓을 안전하게 보호할 수 있으며 데이터의 손실을 일으키는 먼지나 오염물질들이 묻지 않게 된다.

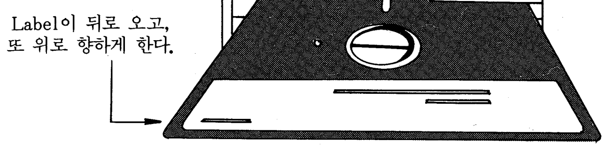
적당한 온도범위에서 디스켓을 보관하라. 지나친 저온(10℃ 이하)이나 고온(52℃ 이상)에서는 디스켓에 들어있는 데이터가 손실될 염려가 있다. 지나친 열은 디스켓을 변형시킨다.

디스켓에 레이블을 꼭 붙이도록 하라. 처음에는 디스켓이 몇개 되지않아 기억할 수 있지만 자료를 모으다 보면 얼마 안있어 많은 수의 디스켓을 갖게 되는데 레이블이 없는 똑같은 디스켓에서 필요한 디스켓을 찾아내기란 무척 어려운 일일 것이다. 이때 레이블이 붙어있다면 모든 디스켓을 로드 해보는 번거러움없이 원하는 내용을 찾을 수 있다. 레이블에 기록하려면 부드럽고 굵은 펜을 사용해야 한다. 그렇지 않을 경우 디스켓에 손상을 입힐 염려가 있다.

### ●주의할 점

민감한 액세스 홀을 손대지 마라. 디스켓은 민감한 자기매질을 입힌 마일러

〔그림 1〕 디스켓 넣는 방법



~なんて、かんたんなものと思っていました。



디스크로 이루어져 있기 때문에, 손가락에 있는 습기가 디스켓 내에 있는 데이터를 디스크드라이브가 리드나 라이트 하는데 방해할 수 있기 때문이다. 디스켓은 내구성의 보호용 자켓으로 싸여 있고 장방형의 액세스 홀을 액세스하도록 되어 있다.

디스켓은 자성을 띠는 물질에서 멀리하라. 만약 디스켓 근처에 자성물질이 있다면 당신의 디스켓을 손상시킬지도 모른다.

또한 디스켓은 음식물이 묻지 않도록 청결히 사용한다. 만약 디스켓에 커피 등을 엎지를 경우 디스켓의 내용이 손상될 수도 있으며 이럴 경우 디스켓을 물기가 있는 상태로 드라이브에 넣는다면 드라이브의 헤드에 치명적인 손상을 입힐 수도 있다.

디스켓을 구부리면 안된다. 구부리면 디스켓내에 접힌 금이 생겨 기록헤드는 데이터를 읽을 때 접힌 금을 건너 뛰어 정보를 읽을 수 없게 된다.

하드디스크는 하드디스크 드라이브라는 캐비넷에 영구히 봉해져 있는 기록매체이다. 이것은 플로피디스크가 갖고 있는 것보다 적어도 수십~수백배 이상의 데이터를 보유할 수 있다.

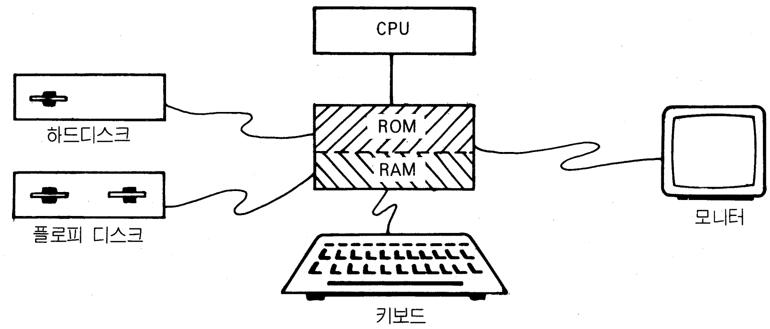
**그림 2**는 퍼스널컴퓨터 시스템의 구성도이다. 중앙처리장치(CPU)는 사람의 '뇌'에 해당되며 연산처리 등의 일련의 작업을 수행한다.

인간의 경우처럼 컴퓨터는 메모리를 갖는데 메모리는 두가지 형태가 있다. 롬(ROM)은 '비휘발성(nonvolatile)' 즉, 컴퓨터를 꺼도 기록된 내용이 그대로 보존된다. 램(RAM)은 보통 '휘발성(volatile)'으로 그 내용이 컴퓨터를 끄면 소멸된다.

## DOS의 기능

DOS는 컴퓨터를 제어하는데 도움을 주는 특수 프로그램의 집합체이다. 이 프로그램들 중에는 키보드로 부터 명령을 받아들이거나, 스크린에 정보를 나타내거나 종이에 정보를 프린트하거나 디스크드라이브를 제어하는 것과 같은 기본적인 작업을 수행하는 것도 있다. DOS는 주로 디스크내에 있는 정보를 관리하는—디스크내에 있는 정보를 분류하고 정보를 읽어 들여 PC로 되돌려 주는—일을 한다. DOS는 그 자체로 워드프로세싱, 스프레드쉬트, 데이터베이스의 개발같은 것은 행하지 않는다. 이런 작업은 응용 프로그램을 사용해서 수행시킬 수 있는데, 컴퓨터의 메모리내에 존재하는 DOS 프로그램없이 응용 프로그램을 사용할 수는 없으므로 시스템을 켤때 메모리로 DOS를 로드시키는 과정이 필요하다. 이 과정을 '시스템 부팅' 또는 'DOS부팅'이라고 한다.

〔그림 2〕 PC시스템 부품 구성도




DOS 명령어로 들어가는 데는 표준형식이 필요한데 이 명령어는 명령어 이름으로 시작해서 차례대로 파라미터를 나열해 간다. 각 파라미터들은 보통 한 디스크화일 또는 디스크화일의 그룹 이름이다.

예를 들면 다음 명령은 MYDATA란 화일을 MYBACKUP이란 화일명 으로 카피하는 것이다.

```
COPY MYDATA MYBACKUP
```

명령을 타자한 다음, ENTER키를 눌러 DOS가 그 명령을 받아들여 수행하게 한다. 그러면 MYDATA란 화일과 똑같은 내용을 MYBACKUP이란 화일명으로 보존할 수 있다. 이렇게 같은 내용을 다른 화일명으로 옮겨놓는 것은 만약의 경우 데이터를 잘못 건드려 지우거나 할 경우를 대비해서이다. 이렇게 하면 언제라도 데이터를 복원할 수 있다.

## 시스템 부팅

앞에서 설명한 바와 같이 시스템 부팅은 컴퓨터를 켜고 DOS를 동작시키는 일이고 시스템을 켤때마다 필요하다. 만일 프로그램내의 버그로 키보드를 동작못하게 되면 시스템을 재부팅시켜야 한다.

□ 주의

만일 하드디스크 시스템을 사용하고 있다면 시작하기 전에 하드디스크가 포맷되어 있는지 확인한다.

일반적으로 하드디스크는 판매자가 시스템을 이니셜 셋업(Initial Set Up) 시키기때문에 사용자가 시스템을 구입하기 전에 이미 되어 있다.

시스템을 부팅하기 위해 DOS시스템 디스켓을 디스크드라이브 A에 넣고 전원을 켜면 다음과 같은 화면이 나온다.

```
Current date is Tue  1-01-1980
Enter new date:
```

MM-DD-YY 형식으로 현재의 날짜를 입력하면 DOS는 현재의 시간을 기억하게 된다. 시간은 HH : MM 형식으로 입력하는데 시간은 0에서 24사이의 정수이고 시간이 입력되면 다음 화면이 나타난다(**화면 1**).

〔화면 1〕

```
Current date is Tue  1-01-1980
Enter new date (mm-dd-yy): 10-05-88
Current time is  0:00:33.77
Enter new time: 5:50


The IBM Personal Computer DOS
Version 3.30 (C)Copyright International Business Machines Corp 1981, 1987
             (C)Copyright Microsoft Corp 1981, 1986

A>
```

□ 주의

날짜 및 요일과 시간 프롬프트를 입력하지 않으면 화일은 DOS버전에 있는 날짜 및 요일과 시간을 그대로 갖게 된다.

화일이 세이브될 때 DOS는 항상 날짜 및 요일과 시간을 기입하기 때문에 만약 사용자가 입력하지 않을 경우는 자동 입력을 호출한다. DOS에 있는 날짜 및 요일과 시간을 그대로 사용하게 되면 언제 화일이 생성되었고, 마지막 수정이 언제 되었는지 알 수 없게 된다. 〈ENTER〉 키를 눌러 날짜 및 요일과 시간 프롬프트를 그대로 통과시키면 다음 화면이 나온다(**화면 2**).

부팅 과정에서 날짜 및 요일과 시간을 입력하지 않을 경우 날짜는 DOS에 들어있는 내용 그대로 체크된다. 따라서 이렇게 될 경우 나중에 화일들을 편집하거나 정리할 때 불편하므로 되도록이면 날짜 등을 입력하는 습관을 들이는 것이 좋다.

다음 **화면 4**의 화면은 날짜와 시간 등이 입력된 경우이다.

**화면 3**과 4에서 본 두 화일블록은 이용성면에서 아주 다르다. 날짜 및 요일과 시간을 입력해야 하는 번거로움을 피하려면 시스템에 리얼타임 클럭을 장치하면 된다. 클럭이 내장되어 있으면 데이터를 세이브시키거나 할때 자동적으로 날짜 및 요일 등이 기록된다.

날짜 및 요일과 시간을 입력하는 일이 끝나면 DOS는 운영 시스템이 사용자가 명령을 내릴 때까지 기다린다는 프롬프트를 나타낸다.

플로피시스템으로 DOS를 부팅시킬 때는 다음과 같이 프롬프트가 나타난다. 이것은 플로피시스템이 한대일 경우는 A로 두대일 경우는 A, B로 지정되기 때문이다.

```
A>
```

하드디스크 시스템에서는 다음과 같은 프롬프트가 나타난다. 하드디스크는 플로피와 마찬가지로 한대일 경우는 C로, 두대일 경우는 C, D로 지정된다.

```
C>
```

프롬프트는 어떤 드라이브가 동작하고 있는가 나타낸다. 다시 말해 DOS는 자동적으로 화일이나 명령을 찾아가고 디폴트 드라이브 상태는 따로 지정하지 않는한 현재 사용되고 있는 상태를 가르킨다. 디폴트 드라이브를 바꾸려면 드라이브 이름을 지정하고 그 다음에 연달아 콜론(:)을 타입한 후 〈ENTER〉 키를 누르면 된다.

듀얼플로피 시스템에서는

```
A>B:
B>
```

하드디스크 시스템에서는

```
C>A:
A>
```

과 같이 한다.

□ 주의

플로피시스템을 부팅했을 때 제대로 부트되지 않으면 드라이브 A에 불량 디스켓이 들어있거나, 디스켓이 없거나, 혹은 DOS 디스켓이 뭔가 잘못된 경우이다.

드라이브 A에 있는 디스켓을 조사해서 DOS 디스켓이 아니라면 시스템을 끄고 다시 시작한다. 만일 DOS 디스켓일 경우 제대로 부트되지 않을 경우는 시스템 니스켓이 불량일 경우이므로 다른 디스켓으로 바꾼다.

□ 주의

플로피 시스템이 디스크 에러라는 메시지를 나타내면 드라이브 A에 불량 디스켓이 있는 것이다.

〔화면 2〕

```
Current date is Tue  1-01-1980
Enter new date (mm-dd-yy):
Current time is  0:00:47.40
Enter new time:


The IBM Personal Computer DOS
Version 3.30 (C)Copyright International Business Machines Corp 1981, 1987
             (C)Copyright Microsoft Corp 1981, 1986

A>
```

〔화면 3〕

```
A>DIR

Volume in drive A has no label Directory of A:\

FILE1    WKS     1280   1-01-80  12:05a
FIEL2    WKS     4210   1-01-80  12:05a
FILE3    WKS    11264   1-01-80  12:05a
FILE4    WKS     3652   1-01-80  12:05a
FILE5    WKS    14658   1-01-80  12:05a

     5 File(s)    326196 bytes  free
```

〔화면 4〕

```
A>DIR

Volume in drive A has no label
Directory of A:\

FILE1    WKS     2115   4-07-85   2:36p
FILE2    WKS    18442   7-23-84   4:53p
FILE3    WKS    41283   7-12-84   2:03p
FILE4    WKS    66270   4-10-85   2:37p
FILE5    WKS    10280   4-15-85   5:48p
MYFILE   WKS    22860   4-24-85  14:22p

      6 File(s)   200010  bytes free
```

다음 **화면 5**와 같은 메시지가 나타나면 DOS 디스켓안에 시스템이 빠져 있거나 불량 디스켓이 사용되었다는 것을 의미한다.

**□ 주의**

DOS 디스켓을 넣은 다음에 안전하게 드라이브 A의 문을 닫지 않으면 시스템이 부팅되지 않는다.

이 경우 드라이브 A의 빨간불이 계속 켜져 있는 상태가 되고, 이것은 드라이브가 작동되고 있다는 것을 의미한다. 이럴때 DOS 디스켓을 빼면 디스켓이 손상될 수 있으므로 컴퓨터를 끄고 DOS 디스켓을 다시 넣고 드라이브 A의 문을 안전하게 닫는다. 그 다음 전원을 다시 넣으면 시스템은 정확히 새로 부팅된다.

**□ 주의**

빨간불은 드라이브가 동작하고 있는 것을 나타내는 것이기 때문에 드라이브내에 디스켓을 넣거나 빼내면 안된다.

〔화면 5〕

```
Non-system disk or disk error
Replace and strike any key when ready
```

이런 경고에 주의하지 않으면 디스켓에 영구적인 손상을 줄 수 있다. 녹음중에 어느 누가 카세트테이프를 빼낼려고 하겠는가?

부팅했을때 화면이 공백상태가 되면 다음 여러가지 문제중의 한 상태이다.

이 경우 맨처음 모니터를 조사한다.

- 코드가 끼워졌는가?
- 켜진 상태인가?
- 밝기 및 명암의 노브가 정확히 조절되었는가?
- 모니터와 컴퓨터 사이의 케이블을 조사했을 때 안전한가? 적당한 출력 잭에 연결되었는가?

이런 조사를 하는 동안 한 두번정도 시스템을 끄고 약 15초 기다린 후 전원을 다시 넣어 시스템을 작동시킨다.

계속 화면이 공백상태이면 전문가에게 의뢰하여 해결하도록 하자.

## 포맷과 카피

DOS 디스켓(카피 가능한 프로그램)을 사용하기 전에 매일 사용하려면 복사본을 만들어 원본을 보관하는 것이 좋다.

포맷팅과 카피에 대해 알기 전에 화일에 관하여 알아보자. 정보는 화일상태로 디스크내에 저장되는데 화일은 데이터나 프로그램으로 구성된다. 텍스트 문이나 정보교환 레코드 등이 데이터 화일의 예이고, 워드프로세싱, 스프레드 쉬트, 데이터베이스 프로그램 등이 프로그램 화일의 예이다. 빈 디스켓을 사용하기 전에 먼저 FORMAT 명령으로 디스켓을 포맷시켜야 한다. 이 명령은 디스켓내의 물리적인 손상을 조사하고 디렉토리를 생성하여 DOS가 화일이 어디에 있는지 알게 한다. 일단 디스켓이 포맷되면 정보를 저장할 수 있다(이전에 어떤 작업을 행한 디스켓—예를 들면 프로그램 디스켓—은 이미 포맷되어 있는데 이 상태에서 다시 포맷하게 되면 디스켓에 포함된 모든 내용이 지워진다).

시스템을 부팅시키는 DOS의 복사본을 만들기 위해서는 먼저 빈 디스켓을 화면 6과 같이 /S 옵션을 사용하여 시스템디스크로 포맷시켜야 한다.

이 과정은 디스크를 포맷시킬 뿐만 아니라 디스크내에 두개의 시스템 화일(히든 화일로서 화면상에서는 나타나지 않는다)을 복사시킨다.

□ 주의

FORMAT 명령은 잠재적으로 위험한 명령이다. 조그만 실수라도 플로피디스크 또는 하드디스크내에 저장된 모든 정보를 지워버린다.

FORMAT 명령을 사용할 때는 정확히 드라이브 명을 지정해야 한다. 명령을 일단 내리면 디스켓내에 있는 모든 내용을 지워버리기 때문에 일단 지워진 내용은 영구히 복구할 수 없다.

이제 포맷이 끝났으면 시스템을 카피하기 위해 COPY 명령을 사용할 수 있다.

COPY 명령을 사용할때 화일별로 따로 따로 카피하지 않고 같은 형태의 것은 묶어서 함께 카피할 수 있는데 이 방법은 와일드 카드로 사용할 수 있다.

와일드 카드는 ＊와 ?가 있다.

＊는 화일명 또는 확장명 대신 사용된다. 예를 들어 ＊.COM이라면 확장명이 COM인 화일 모두를 나타내고 WS.＊라면 화일명이 WS인 모든 화일을 나타내는 것이다.

?는 애매모호한 하나의 문자를 대신하는데 예를 들어 EXAMPLE.××?이면 확장명의 세번째 문자에는 관계없이 다른 글자만 같으면 모두 화면에 출력한다. NIG??.×××이면 화일명은 다섯자이고 마지막 두자에는 관계없이 나타낸다. 이것은 화일명을 확실히 모른다든지 같은 형태의 것을 전부 알아보거나 할때 사용된다.

다음 명령은 화일명 및 확장자에 관계없이 모든 화일을 카피한다.

```
A>COPY A:*.* B:
COMMAND.COM
ANSI.SYS
FORMAT.COM
CHKDSK.COM
SYS.COM
DISKCOPY.COM
DISKCOMP.COM
COMP.COM
EDLIN.COM
MODE.COM
FDISK.COM
BACKUP.COM
RESTORE.COM
PRINT.COM
RECOVER.COM
ASSIGN.COM
TREE.COM
GRAPHICS.COM
SORT.EXE
FIND.EXE
MORE.COM
BASIC.COM
BASICA.COM
        23 File(s) copied
```

듀얼 플로피시스템에서 COPY 명령을 사용하려면 먼저 드라이브 A에 원본 디스켓(DOS 시스템디스켓)을 넣고 시스템을 부팅시킨다. 시스템이 부팅된 후 커서가 나타나면 드라이브 B에 복사하고자 하는 디스켓을 넣고 드라이브 A에서 복사하고자 하는 화일을 B드라이브로 카피한다. 이때 원하는 화일만 카피할 경우는 A>COPY 화일명.B:과 같이 한다.

전체를 카피하고자 하면 다음의 화면 7과 같이 한다.

□ 주의

DOS 매뉴얼에 있는 DISKCOPY 명령을 사용하면 백업을 시킬수 있다. 이 방법은 신속하고 쉽게 정확한 카피를 하지만 잠재적인 문제에 조심해야 한다.

DISKCOPY는 한 디스켓에서 모든 내용을 다른 디스켓으로 카피하는 명령이다. 이때 카피되는 디스켓이 포맷되지 않았다면 카피되는 동안 포맷된다. 그러나 만일 카피할 내용이 들어있는 디스켓이나 카피되고 있는 디스켓에 불량 섹터가 존재하면 제대로 카피할 수 없다. 또한 DISKCOPY 명령은 플로피디스크에서 플로피디스크로 카피하고자 할 때만 사용이 가능하다.

〔화면 6〕

```
A>FORMAT B:/S

Insert new diskette for drive B: and strike
any key when ready

Formatting...Format complete System transferred

      362496  bytes total disk space
      40960  bytes used by system
      321536  bytes available on disk

Format another (Y/N)?N
```

〔화면 7〕

```
A>COPY *.* B:

COMMAND.COM
      :
ANSI.SYS
      :
      :
23 File(s) copied
```

# 전자 · 통신 · 기계분야
# 통신교육과정 신설안내

각 기업체에서는 기업 생존전략으로서 종업원에 대한 교육의 일환으로, 보다 효과적이며 생산적인 통신교육방법을 채택하여 종업원의 교육훈련에 많은 투자를 하고 있읍니다. 그러나 현재 국내에서 실시되고 있는 통신교육과정은 대부분 사무와 경영의 관리면에 편중해 있어 급격히 발전되고 있는 첨단산업에 보다 능동적이고 현명하게 대처하기 위해서는 그 방면에 효과적인 통신교육 교재 개발이 절실하게 요청되고 있는 바, 본사에서는 산업계 엔지니어의 자기 개발을 위해 새로운 기술과정을 중점적으로 하여 첨단산업 및 전자 산업분야에 아래와 같은 새로운 통신교육 교과과정을 개설하게 되었으니 관계되시는 분이나 관심이 있으신 분은 많은 참여를 바랍니다.

| 수강과목 | 수강기간 | 수 강 대 상 자 | 매월별 교과과정 항목 |
|---|---|---|---|
| **1989년 2월 개설** | | | |
| IC기술 기초 | 6개월 | 민생기기, 전자기기, 제어기기, 전기·전자부품 재료 등 분야의 연구 개발, 설계기술, 생산기술자 | ① IC의 기초 ② 개별 반도체 ③ 아날로그 IC의 기본과 응용 ④ 디지탈 IC의 기초 ⑤ 디지탈 IC 회로 ⑥ 마이컴시스템의 기초 |
| IC응용설계기술 | 6개월 | 민생기기, 전자기기, 제어기기, 전기·전자부품 재료 분야의 연구 개발, 설계기술, 생산기술자 및 IC기술 기초과정 수료자 | ① 반도체의 기술 ② IC 주변기술 ③ 아날로그 IC 응용회로 ④ 센서와 IC 응용회로 ⑤ 디지탈회로 ⑥ 디지탈 IC 응용 |
| 메카트로닉스 입문 | 6개월 | 생산기술, 연구, 개발, 설계, 제조, 설비보전, 영업기술 등 각 분야의 기계기술자, 설비의 자동화를 위해 메카트로닉스의 지식이 필요한 자 | ① 기계와 일렉트로닉스의 기본 지식 ② 센서 선택법과 연결법 ③ 액튜에이터 선택법과 연결법 ④ 콘트롤러의 역할과 마이컴제어의 고찰방법 ⑤ 메카트로닉스의 실제(1)－마이컴에 의한 모터제어 ⑥ 메카트로닉스의 실제(2)－현장에서의 응용사례 |
| **1989년 6월 개설** | | | |
| 노이즈대책 실무 | 6개월 | 전자기기, 제품의 연구 개발에 필수인 회로설계 및 실장설계 기술자, 생산설비, 검사, 계측에 종사하는 생산기술자, 제조 가공 기술자 | ① 노이즈대책의 기본 ② 노이즈에 강한 실장설계 ③ 노이즈에 강한 아날로그회로 설계 ④ 마이컴 시스템의 노이즈대책 ⑤ 산업용 기기에서의 노이즈대책 사례 ⑥ 노이즈 측정법과 각종 규격 |
| 센서 주변회로 설계 | 6개월 | 전기·전자기술자, 메카트로닉스 기술자 | ① 센서의 기초와 주변기술 ② 온도센서 ③ 광센서 ④ 자기센서 ⑤ 기계량 센서(길이, 압력, 위치, 기타) ⑥ 가스 온도센서 |
| 일렉트로닉스 제어 기초 | 6개월 | 연구 개발, 생산기술, 설계, 제조, 설비보전, 영업기술 각 분야의 설계기술자, 강전계 기술자 | ① 일렉트로닉스 제어의 기초 ② 디지탈 IC의 기본회로 ③ 센서 응용과 디지탈 입출력 ④ 파워 베이스의 기초와 응용 ⑤ 프린터 기본설계와 실장설계 ⑥ PC(프로그래머블 콘트롤러)와 기계제어로의 응용 |

**개설 예정과목 및 수강대상자**

- **아날로그 응용기술** : 제품 개발 및 생산설비 개발의 회로설계에 종사하는 자
- **아날로그 처리기술** : 연구 개발부문, 생산기술부문에서 회로설계에 종사하는 전자·전기기술자, 메카트로닉스 기술자
- **데이터통신 입문** : 시스템 엔지니어, 소프트웨어 기술자, 전산실 요원, 생산기술자, 제조기술자, 검사담당자, 영업담당자, 기타 현 업무가 데이터통신기술과 관련있는 자
- **데이터통신기술** : 통신 단말의 인터페이스 설계자, 소규모 LAN 개발자, 통신기술부문의 초급자
- **퍼스컴통신기술** : OA, FA, HA, LA, SA 등의 시스템 설계, 개발기술자, 생산기술자, 퍼스컴을 이용한 데이터통신 기술자
- **디지탈신호처리 기술** : 기기 개발, 회로설계에 관계되는 연구, 개발 설계자, 시스템 설계자, 서비스 엔지니어, 생산기술자
- **디지탈신호처리기술** : 기기장치의 개발 또는 회로설계에 관한 연구 개발, 설계부문의 기술자

**통신교육 상담 문의안내**

주소 : 서울시 용산구 갈월동 17번지
전화 : (02)754－1044(代)
FAX : (02)797－1049

# 국내소식

—과기처
## 국책연구 심의위원회 운영

정부는 특정연구 개발사업의 효율적 추진을 위해 과기처장관이 주재하고 관련부처 및 학계·산업계·연구계·전문가들이 공동참여하는 국책연구개발사업 심의위원회를 구성, 운영하기로 했다.

과기처에 따르면 첨단기술을 중심으로한 연구개발의 복합화·거대화 추세에 대응, 이 심의위원회를 통해 산·학·연 연구능력을 결집시켜 나가기로 했다.

동 위원회는 과기처장관을 위원장으로 각계에서 30인 이내로 위원회를 구성, 국가연구개발 중장기 계획을 수립하고 주요 국책연구개발사업을 심의, 조정하게 된다.

또 연도별 사업기본계획 수립 및 중요정책 심의, 중점연구분야 설정, 주요 연구과제 최종심의 등의 기능도 갖는다.

과기처는 특정연구과제를 선정, 심의하는 기구인 분야별 실무위원회를 9개에서 12개로 늘려 전문성과 공정성을 기하기로 했다.

—체신부
## INFO SEOUL '88세미나 개최




지난 11월 2일부터 6일까지 잠실 롯데월드에서는 데이터베이스를 비롯한 각종 정보서비스를 주제로 INFO SEOUL '88 국제세미나 및 전시회가 개최되었다.

체신부가 주관한 이번 행사에서는 매일경제신문사를 비롯한 국내 39개 데이터베이스 퍼스컴 모델생산업체와 일본 마루젠 등 외국업체 8사를 포함, 모두 47개 업체가 참가하였다.

이번 전시회에는 하드웨어 업체보다 소프트웨어업체가 많이 참가한 것이 특징으로, 소프트웨어업체가 출품한 내용은 크게 경제정보·증권정보·기업정보·기술정보 등 5개 분야로 나뉘어 전시되었다.

또한 하드웨어 출품업체로는 금성반도체, 삼성전자, 현대전자 등이 참가하였다.

한편, 이번 행사는 세계 각국의 통신주관청, 정책당국자, 공중통신사업자, 정보서비스 제공자 등이 대거 참여함으로써 각국의 정보산업정책 및 기술정보의 교류와 국제간의 공동사업추진을 모색하는 등 우리나라 정보산업의 발전은 물론 국제적 지위를 확보할 수 있는 계기가 되었다.

—체신부
## 국민관광 예약시스템 확대적용

체신부는 국민들의 여행 관광편의를 위해 한국데이타통신이 지난 8월부터 제주지역을 대상으로 운영하고 있는 국민관광 예약시스템 적용지역을 12월부터 경주, 설악산 지역의 호텔 콘도미니엄, 골프장, 여행사에까지 확대하기로 했다.

또한 이용자의 수요가 많을 경우에 내년부터 단계적으로 확대할 계획인 이 시스템은 일종의 VAN(부가가치 통신망)으로 데이타통신에 설치된 호스트컴퓨터에 유명관광지의 숙박업소, 레저업소 등 관광정보를 입력하여 전국의 가입 여행사 터미널과 연결된 컴퓨터망으

로 이용자가 가입 여행사의 터미널을 통해 여행하고 싶은 지역의 관광업소, 정보안내, 예약, 발권 등의 서비스를 받을 수 있게 되어 있다.

-체신부
## 우편물 배달서비스 확대 실시

체신부에서는 현재 대도시 21개국에서 운영중인 주소 이전 우편물의 컴퓨터추적 배달서비스를 31개국으로 확대 실시키로 하였다.

컴퓨터추적 배달서비스란 이주시 주민등록 퇴거후 주소 이전 신고서를 우체국에 제출하면 우체국에서 이를 컴퓨터에 입력, 기존 주소로 오는 우편물을 추적 배달해 주는 서비스를 말한다.

-체신부
## 행정전산망용 주전산기 전담

그동안 논란이 많았던 국산 행정전산망용 주전산기의 운영 및 책임을 앞으로 체신부가 전담케 됐다.

지난 10월26일, 한국 컴퓨터연구조합 주최로 힐튼호텔에서 열린 「행정전산망용 국산 주전산기 발표회」에서 오명 체신부장관이 발표한 바에 의하면, 그동안 상공부, 과기처, 체신부 등 3개부처에 분산돼 혼선을 빚었을 뿐만 아니라, 연구개발이 지연되어 오다가 이번 국무회의에서 전담기관을 체신부로 결정함에 따라 이러한 문제점들을 극복하게 된 것이다.

한편 이번에 국산화한 행정전산망용 주전산기는 올해 18대가 주민·토지관리업무 전산화용으로 공급될 예정이고 내년까지 모두 1백42대가 행정전산화용으로 생산될 예정이다.

-한국전기통신공사
## 운용시스템 통합관리체제로

한국전기통신공사는 현행 개별 통신망 운용보전 시스템이 적용분야와 사용기술에 따라 독립 관리돼 기능 중복은 물론 전체적인 망관리가 불가능하기 때문에 ISDN(종합정보통신망) 서비스에 부적당하다고 판단하고, 2천년대초 ISDN의 실용화에 대비하여 현재의 운영보전시스템을 내년부터 단계적으로 통합관리체제로 전환할 계획이다.

이 계획이 진행됨에 따라 망 설비의 표준화가 가능해져서 기능적으로 분산된 통신망을 이용하여 음성, 데이터 영상 등 모든 정보교환이 가능하게 된다.

이를 위해 전기통신공사는 1단계가 시작되는 내년에 기존의 운용시스템의 기능확장 및 개선작업과 분야별 운용보전센터를 설치한 다음, 90년부터 ISDN 교환기 운용보전시스템 개발에 착수하고, 90년대 중반까지 정리될 ISDN 운용보전 시스템과 관련된 원격검침기술, 망접속기술, 통신망, 데이터관리기술, 인공지능기술, 컴퓨터 응용기술, 상황판단기술 등을 내년부터 확보해 나갈 방침이다.

현재의 운용보전 시스템은 CSMS, TLMOS, SLMOS, AOM, CTMS, CAROT 등 6개 분야가 있다.

-한국전기통신공사
## 90년대까지 통신시설 디지탈화

한국전기통신공사는 현재 추진 중인 통신시설의 디지탈화를 오는 90년대 중반까지 마무리하여 빠르면 96년부터 모든 통신서비스를 고품질의 디지탈 장비로 제공키로 하였다.

동사에서는 통신서비스의 선진화를 위해서는 교환기 및 전송로 등 주요시설의 디지탈화가 시급하다고 판단하고, 90년대 중반까지 모든 전화국용 국설교환기와 기간전송로를 디지탈 시설로 교환키로 하였다.

교환기의 경우, 서울·부산 등 대도시 전화국은 시내와 시외 겸용이 가능한 10만회선 규모의 디지탈 교환기를 설치하고, 탠덤중계방식을 채택하여 2개 이상의 전화국에서 오는 중계선을 수용, 이를 상호간에 교환 접속하도록 했다. 또한 농어촌 지역의 교환시설은 2만회선 규모의 중용량 디지탈 교환기를 설치할 예정이다.

그리고 새로 가설되는 전송로의 경우에는 시단위 이상은 비음성 서비스도 가능한 광케이블을 설치할 계획인데, 시설용량은 앞으로 15년후의 수요까지를 감안하여 최대단위로 결정하되, 서울~부산간 등 기간전송로는 한 회선이 고장나더라도 다른 회선을 통해 전송이 가능하도록 자동절체기능을 갖추기로 했다.

-한국전기통신공사
## 내년부터 비디오텍스 서비스 제공키로

한국전기통신공사는 공중전기 통신망을 통한 비음성서비스 활성화 시책의 일환으로 내년부터 비디오텍스 서비스를 제공키로 했다고 관련업계는 발표하였다.

동사는 이미 지난 6월말경 이같은 서비스에 필요한 비디오텍스 서비스를 선정하기 위해 두산·삼성HP·현대전자·IBM 등 4개업체를 대상으로 제안설명회를 가졌으며 최근 현대의 하드웨어 「피라미드9840」과 소프트웨어 「VCS-DB」를 내정하였다.

통신공사는 연내에 비디오텍스 시스템을 도입, 일단 10대의 단말기를 접속, 내부 시험을 통해 기능보완 및 각종 DB와의 인터페이스 장치개발을 마친뒤 내년부터 2백대 가량의 단말기로 시범서비스를 실시할 계획이다.

초기단계에서는 증권사 등이 미리 구축해 놓은 증권 DB를 이용, 증권시세 안내서비스와 114안내정보 등을 제공하고 점차 각종 공공정보 및 생활정보를 추가해 나갈 계획이다.

-한국정보산업연합회
## 프로그램 등록 관련업무 맡아

지난해 9월부터 과학기술처가 담당해 왔던 컴퓨터 프로그램 보호법 규정에 의한 프로그램등록 관련업무가 지난 11월 1일부터 한국정보산업연합회에 위탁, 수행케 되었다.

연합회에서 담당 수행하게 된 등록업무로는 프로그램등록, 프로그램 저작권 등록, 프로그램등록 및 프로그램 저작권등록 사항 변경, 프로그램 등록부 및 프로그램 저작권 등록부 열람, 등록부 사본교부 등이며, 프로그램복제물 보호만은 50년동안 장기간 보존하여야 하고, 동시에 소홀히 할 수 없는 중요성을 감안하여 당분간은 과기처에서 계속 담당키로 하였다.

이처럼 과학기술처가 한국정보산업연합회에 동 등록업무를 위탁하게 된 것은 대민봉사 강화와 프로그램 등록과 관련한 홍보 등 제반사항을 적극 개발 추진코자 하는 정책적 의도에서 기인한 것이라고 할 수 있다.

–한국정보산업연합회

## S/W 개발촉진법 설명회 가져

사단법인 한국정보산업연합회는 지난 10월 17일 여의도 전경련 대회의실에서 '소프트웨어 개발 촉진법. 령 종합설명회'를 가졌다.

이날 설명회는 장경철(과기처 정보산업과)과장이 강사로 나와 입법의 배경 및 경위, 법의 성격과 운영방침, 그리고 촉진법 시행령 각 조문별 내용을 2시간에 걸쳐 설명한 후 청취자의 질문을 받는 순서로 진행됐다.

동 촉진법은 지난 86년, 소프트웨어에 대한 지적소유권 보호제도 도입에 따른 국내소프트웨어 산업의 보호, 육성을 목적으로 하며 초기단계에 있는 국내소프트웨어 개발능력을 결집하고 소프트웨어의 고유영역 구축 및 국제시장성 창출이라는 기대효과를 갖고 있다.

특히 동법과 같이 시행될 '소프트웨어 개발비 산정기준'은 타제품에 비해 물리적인 가치측정이 어렵고, 그 규모 및 성능에 있어서 특수한 성격을 지닌 소프트웨어의 개발비, 그리고 생산업자를 위한 각종 입찰, 품질보증, 유지보수 등 제반 공제사업에도 크게 기여할 것으로 보고 있다.

–경한시스템(주)

## 심볼 및 로고 갱신

경한시스템(주)(대표 : 최금영)는 지난 10월 1일자로 새로운 경한이미지를 부각시키고자 기존의 심볼 및 로고타입을 갱신하였다.

심볼마크의 모티브는 '경한'의 두문자인 "ㄱ"자에 경한의 의지를 더한 것으로, 약동하고 힘차게 전진하는 의지를 형상화하였다.

–경한시스템

## 원주의료원에 전산시스템 설치

경한시스템(사장:최금영)은 전국 31개 지방공사 시, 도 의료원의 전산화를 위한 컨버전트 시스템을 현재 정상 가동중이다.

이 시스템은 동사가 지난 4월 6일 (사)전국지방공사 의료원연합회의 설치계약을 체결한 이후 각 시·도별로 200병상 이상의 규모에는 시스템S/320을, 그 미만의 규모에는 시스템S/221을 설치하도록 계획을 세우고 4월 중순경부터 시스템 설치에 따르는 제반 교육실시 및 부대시설을 설치하는 등 체계적인 준비를 통해 추진된 것이다.

또한 이 시스템은 6월초에 원주, 의정부의료원의 설치를 시작으로 조직적이고 체계적인 업무진행 및 뛰어난 기동성을 발휘하게 되었으며 시범의료원으로는 원주의료원이 지정되었다.

–금성사

## 텔리텍스트수신기 개발

금성사는 차세대 문자다중방송 방식으로 각광받고 있는 플로프방식의 텔리텍스트 수신기를 개발, 내년초부터 수출에 나설 계획이다.

현재의 텔리텍스트 방식은 WST로 불리는 영국·서독방식(UK텔리텍스트)이 세계의 95%를 점유하고 있는데 이 방식은 정보를 시간에 따라 순서대로 송신, 정보를 얻기까지 30여초를 기다려야 하는 등 수신시간 지연현상이 단점으로 지적되어 왔다. 반면 금성사가 개발한 텔리텍스트 수신기는 수신시간 지연현상을 제거한 새로운 문자다중 방식이다.

한편 플로프 텔리텍스트 수신기는 89년부터 영국, 스웨덴 등 북부유럽 에서 선보일 예정인 플로프방식의 문자다중 방식을 수신할 수 있으며, 특히 컴퓨터의 기억용량을 최대 1백28페이지까지 늘려 정보를 미리 기억시켜 두었다가, 원할 때 즉시 확인할 수 있도록 한 첨단기기이다.

또한 금성사는 이미 UK 텔리텍스트 수신기를 개발해, 올해 유럽지역에 3만여대 수출한 바 있으며, 이번 텔리텍스트 개발로 인해 유럽시장 진출이 가능해졌다.

–대우통신

## 반도체 디자인센터 설립

대우통신(대표:박성규)은 지난 10월 17일 미국의 ZYMOS사와 반도체 디자인센터 설립계약을 체결하였다.

동 센터는 박성규 현 대우통신 사장을 대표이사로 하고 대우통신측 3명, ZYMOS측 2명 등 총 5명으로 경영진이 구성되었다.

동사는 ASIC(주문형 반도체)분야를 전략사업으로 결정하고 지난 86년 3월, 반도체 사업에 착수한 이래 ZYMOS사로부터 기술습득 및 자체 기술력을 확보해 왔는데 현재 3$\mu$m~5$\mu$m급의 바이폴러와 CMOS의 제조공정 및 일부 설계 기술을 보유하고 있다.

또한 금년 상반기에는 ZYMOS사로부터 CMOS 생산설비를 도입하고 본격

적인 양산체제에 돌입하였으며, 최신 자동설계기술의 조기습득 및 시장확보를 위해 동 센터를 설립하게 된 것이다.

동사는 현재 1.8$\mu$m 선폭의 표준 셀 방식을 위주로 ASIC을 설계 생산하고 있으나 국제 경쟁력을 높이고 집적도를 높이기 위하여 1.2$\mu$m 선폭의 표준 셀 라이브러리를 갖출 계획이며 이를 위해 기술인력확보 및 최신의 디자인 설계툴 개발에 역점을 두고 아울러 BI-CMOS 게이트어레이 및 PLD분야에도 진출하는 등 사업영역을 점차 확대해 갈 방침이다.

전화 : (02)559—2604
• 장소 : 서울시 서초구 서초동 1318-2 기남B/D 10층 삼성전자OA 고객센터내 고객교육실.

| 구 분 | 기 종 | 과정명 | 11월 | | | | | 12월 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 |
| 16비트 | SPC-3000A | 기본과정 | | | | ○ | | | ○ | | |
| | | MY-워드 | | | | ○ | | ○ | | | |
| | | | | ○ | ○ | | | | | ○ | |
| | | DBASE Ⅲ PLUS | | ○ | | | | | ○ | | |
| | | 셈벗-16 | | | ○ | | | | | ○ | |
| | SPC-5000 | XENIX | | | ○ | | | | | ○ | |
| 32비트 | SPC-8000 | UNIX | | ○ | | | | | ○ | | |
| 8비트 | SPC-1500A | 학생교실 | | ○ | | | | | ○ | | |

–대원시스템
## 주식투자지원시스템 개발, 시판

과학적인 투자 분석기법을 활용, 주식투자가들에게 정확한 주식투자정보를 제공해주는 SIS-Ⅱ시스템을 대원시스템(대표:박진만)에서 개발, 시판에 나섰다.

16비트 퍼스컴용으로 개발된 이 시스템은 메뉴선택방식을 채택하여 컴퓨터에 대한 기초지식이 없는 초보자도 쉽게 활용할 수 있다.

또한, 시·고·저·종가 및 거래량 자료를 신속하게 자동 입력하고 매입·매도 시점과 투자종목을 상세하게 분석할 수 있어 전문 투자가들은 물론 소액 투자가들도 손쉽게 사용할 수 있다.

한편 이 기기는 대용량의 하드디스크 드라이브를 부착, 4백80여개 상장회사들의 주식정보를 신속하게 검색할 수 있다.

–삼성전자
## 4/4분기 컴퓨터 교육일정 발표

삼성전자 정보기기 사업본부에서는 '88년 4/4분기 컴퓨터 교육일정을 발표하였다.

일정 및 문의처, 장소는 다음과 같다.

• 교육내용문의처
삼성전자 정보기기 사업본부

–삼성전자
## 한방용 소프트웨어 발표

삼성전자(주)는 소프트웨어 개발 협력업체인 SSM회원사에서 개발한 "한방 병, 의원전용 컴퓨터시스템" 소프트웨어를 발표하였다.

한방브레인 740이라 명하는 이 소프트웨어의 사용기종은 SPC-1500이며, 개발업체는 삼영컴퓨터이다. 문의처는 다음과 같다.

문의처 : 삼성전자 정보기기(본)S/W과 (02)559-2660~2
삼영컴퓨터(02)419-4744~5

–삼성반도체통신
## 합병후 삼성전자로 출범

지난 11월 1일을 기해 삼성전자가 삼성반도체와 통합, 새 이름인 삼성전자주식회사(Electronics CO., Ltd.)로 출범하였다.

전자업계의 세계적 추세인 산업전자 부분을 강화하기 위해 단행된 이번 통합으로 삼성은 올해 총매출액이 3조7천억원으로 예상하고 있다.

가전·반도체·정보통신 부문을 통합, 새롭게 태어난 동사는 반도체와 통신기술을 핵으로 하여 기술의 복잡화·다기능화·시스템화를 추진, 미래 정보산업을 주도하게 될 것이며, 90년대 중반까지는 단일회사 규모로 세계 10위, 2천년까지는 IBM, 마쓰시다, 히다찌, 도시바와 함께 세계 5대 전자업체로 발돋움할 것을 목표로 하고 있다.

–삼성전관
## 플라즈마 표시판 개발

삼성전관(대표 : 김정배)은 삼성종합기술원과 공동으로 컴퓨터 및 가전제품용 플라즈마 표시판을 개발했다.

미·일·프랑스에 이어 세계 4번째로 삼성에 의해서 개발된 이 제품은 2장의 유리사이에 플라즈마라는 방전가스를 넣어 고압전류에 의해 자체 발광하는 원리를 이용한 표시판으로 컴퓨터의 모니터 및 TV에 사용된다.

또한 이 표시판은 액정표시판에 비해 빛 발산 속도가 빠르고 시야각도가 넓으며 화면이 선명해 최근들어 점차 널리 활용되기 시작한 첨단기술제품이다.

미 IBM이 첫 개발한 이 제품은 선진국들이 기술이전을 기피하는 제품으로 삼성이 이를 자체 개발함으로써 우리나라에서도 고부가가치의 전자제품 생산이 가능하게 되었다.

–쌍용컴퓨터
## FA분야 대폭 강화

쌍용컴퓨터(대표 : 박병철)는 미국 알렌브래들리사와 제휴하여 공장자동화분야의 시스템 통합사업을 크게 강화한다.

쌍용컴퓨터에서는 공장자동화 전문업체인 알렌브래들리사와 시스템 통합에 관한 기술과 장비 및 교육훈련 등 제반 사항에 관해 상호협력관계를 유지키 위한 합의각서를 교환했다고 밝혔다.

이에 따라 쌍용컴퓨터는 지금까지 축적해 온 각종 소프트웨어 개발기술 및 장비를 결합, 그룹내 공장증설에 따른 FA프로젝트와 국내 대형 FA프로젝트에 적극 참여할 예정이다.

### -쌍용컴퓨터
## 전문가 시스템 개발도구 SKI-Ⅱ개발

쌍용컴퓨터는 최근 과학기술원과 지난 87년 8월부터 공동으로 추진해 온 전문가시스템 개발도구인 SKI-Ⅱ(Sangyong KAIST Intelligence)를 국내 최초로 개발, 인공지능(AI)분야의 새로운 장을 열었다.

SKI-Ⅱ는 지난해 개발한 세무자문전문가시스템 개발용 도구인 SKI-Ⅰ을 일반적 자문형태의 전문가시스템 개발도구로 활용할 수 있도록 기능을 다양화시킨 것이다. 따라서 이번에 개발된 SKI-Ⅱ는 상속세법, 독과점규제법, 공정거래법 등 제반 법률에 관한 자문은 물론 기업공개, 해외증권발행요건, 의사결정방법, 경제정책 등의 경영에 관한 자문과 통계분석, 원자력발전기의 이상상태진단 등과 같은 기술정보관리, 여행안내 등과 같은 생활정보시스템 개발 등의 분야에서 활용이 가능한 것이다.

한편 동사는 올해안에 SKI-Ⅲ의 개발을 착수할 예정이다.

### -아폴로컴퓨터
## 분산처리용 NCS IBM에 제공

아폴로컴퓨터는 한국IBM이 분산처리를 위해 동사의 NCS(Network Computing System)을 사용하기로 했다고 발표했다.

아폴로의 NCS는 컴퓨터 네트워크를 통합하고, 응용프로그램을 여러 종류의 컴퓨터에 분산시키는데 사용되는 소프트웨어 툴로써, IBM에서는 이 NCS 테크널로지를 IBM의 AIX 오퍼레이팅 시스템에 도입하여 자체의 DS(Distributed Services)를 향상시킬 예정이다.

NCS의 특징으로는 응용프로그램을 수퍼컴퓨터나 수퍼미니컴퓨터, 미니수퍼컴퓨터, AI엔진, 데이터베이스 메니지먼트 머신, 워크스테이션 등 각 전문시스템에 분배시켜주며, 작업을 마친 시스템에는 네트워크를 통하여 자동적으로 프로그램을 분배해 주므로써, 모든 시스템을 효율적으로 사용할 수 있게 해준다.

1987년 초 아폴로에 의해 소개된 NCS는 교육기관이나 정부기관 및 우주항공, 전자, 금융, 과학연구분야 등 산업전반에 걸쳐 네트워크 컴퓨팅의 산업표준으로 인정받고 있다.

### -왕컴퓨터 코리아
## Wang PC 280의 3270에뮬레이션 카드 개발

왕컴퓨터 코리아에서는 Wang PC 280을 IBM S/370, 43××, 30×× 시스템의 한·영 온라인 터미널(3278)로 사용할 수 있는 3270 에뮬레이션카드와 SNA통신응용 프로그램을 개발하였다.

동 제품의 특성을 살펴보면, 핫키를 이용하여 PC 컴퓨터 모드와 3270에뮬레이션 모드 사이의 변환이 신속하게 이루어지며 리모트 모드뿐만 아니라 로컬 모드에서도 PC280을 IBM 5550, 3278과 함께 호스트 시스템의 터미널로 사용할 수 있다. 또한 PC모드의 화일 전송기능을 이용하여 IBM 대형 시스템과 화일 전송이 간편하게 이루어지며, CICS, TSO, IMS, ROSCO등 호스트 시스템의 유틸리티도 이용할 수 있다.

그리고 3274/3174터미널 제어장치를 통해 호스트 시스템과 프린터를 공유하므로 비용과 설치공간을 절약할 수 있고, ITT의 터미널 제어장치를 사용하는 곳에서도 별도의 설비가 필요없이 간단한 조작으로 연결이 가능하다.

### -익산전자
## 다중통신장치 국산화에 성공

(주)익산전자는 컴퓨터 주변기기인 다중통신장치(ICM호환 멀티플렉서)의 국산화에 성공하였다.

동사는 광고용 컴퓨터관리 전문업체로 1년여에 걸쳐 3억원의 연구비를 투입, 동 시스템을 개발하였는데 이로인해 연간 1백억원 이상의 수입대체 효과를 기대할 수 있게 되었다.

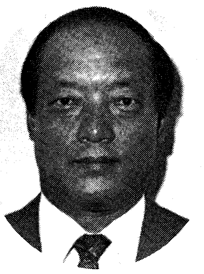
이 제품의 특징은 데이터의 저장·삭제·편집기능 등이 종전의 수입품에 비해 우수할 뿐만 아니라 4배 이상(4백만 단어)의 데이터를 저장할 수 있다. 또한, 동 기기는 기종이나 성능이 서로 다른 컴퓨터, 텔렉스, 팩시밀리, 프린터 등에 연결하여 속도·신호 및 부호·코드 등을 조정하여 통신할 수 있다.

한편, IBM호환 퍼스컴과도 쉽게 연결할 수 있으며, 자료전송속도를 초당 50비트에서 1만9천2백비트까지 자동조절할 수 있다.

익산전자는 3억원을 투입, 경기도 용인공장내에 생산시설을 갖추고 내년 초부터 양산에 들어갈 계획이다.

### -한국 유니시스(주)
## 조완해 신임사장 취임

컴퓨터 전문업체 한국 유니시스(주)는 최근 당사 조직개편에 따라 해리화성 김사장을 회장으로 승진 발령하고, 조완해(趙完海)씨를 신임사장으로 영입하였다.



조완해 신임사장은 경북대 수학과를 졸업하고 지난 1967년 한국 IBM 주식회사에 입사하여 1983년까지 전무이사로, 1987년부터 88년까지 왕컴퓨터코리아(주) 부사장으로 일해 오다가 이번에 한국 유니시스주식회사 사장으로 영입된 정통 컴퓨터맨이다.

### -한국 IBM
## '88 소프트웨어 전시회 개최

한국 IBM(대표 : 피터 더 로울리)은 지난 10월20일과 21일 양일간 커스터머센터에서 88소프트웨어 전시회 및 관련세미나를 개최했다.

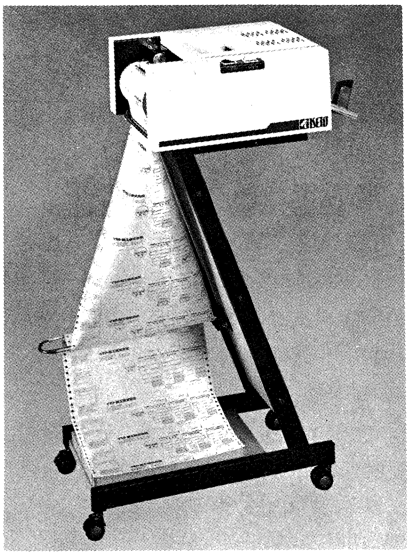
이 행사는 정보처리산업의 필요성을 인식, 전산도입을 고려하고 있는 신설 중소업체의 요구에 부응하여 마련된 것으로, 동사의 생산관리 프로그램인 MAPICS, POS 전산화 프로그램 및 기술/과학분야의 소프트웨어를 비롯하여 관련업체의 다양한 소프트웨어 패키지들이 전시되었다.

또한 전시회와 같이 마련된 세미나에서는 업체에서의 전산시스템 도입 및 설치방안에 필요한 제반사항들을 다루었는데 IBM의 중소형 시스템인 9370, AS/400 및 5363 시스템이 소개됐다.

### -한일비지네스
## DM 발송용 우편자동화 시스템 공급

한일비지네스에서는 일본 이세도사와 대리점 계약을 체결, 국내에 DM 발송용 우편자동화 시스템을 공급한다.

이 시스템은 컴퓨터로 찍혀 나온 각종 청구서 등 우편물을 자동으로 봉합



해 엽서나 봉투형태로 만들어 주는 우편자동화' 기기이다.

봉투봉합 시스템인 메일러(MAILER·모델3100)는 봉투크기를 자유롭게 조절, 2단에 우편내용물 및 4종류까지의 광고물을 삽입, 자동봉합하는 시스템이다. 또한 동 시스템은 봉투 앞면에 창문형태로 수신자 주소가 드러나게 설계돼 있으며, 컴퓨터로 내용물을 분류, 수신자에 따라 서로 다른 우편물을 보낼 수 있도록 돼 있는데, 1시간에 최고 1만2천매의 DM을 처리할 수 있다.

한편, 엽서봉합 시스템인 실러(SEALER·모델5300)는 컴퓨터로 찍혀 나온 청구서를 엽서형태로 접어, 자동봉합해 주는 것으로 1분당 30매까지 처리할 수 있다.

### -효성컴퓨터
## C-ISAM 한글버전 발표

효성컴퓨터에서는 지난 7월, (주)다우와의 협조하에 한글 DBMS인 HISQL(한글 informix-SQL)을 발표한데 이어, 이번에는 informix ESQL-C, informix-4GL, C-ISAM의 한글버전을 발표함으로써 다목적 분산처리 시스템인 POWER NET(386 유닉스 머신)에 완벽한 informix DBMS를 제공할 수 있게 되었다.

HISQL은 표준 데이터 언어인 SQL을 기본으로, 어플리케이션개발을 위하여 필수적인 화면작성, 대화형 에디터, 보고서 출력, 데이터베이스 관리용 유틸리티 등의 기능을 제공한다. 그리고 informix-ESQL/C는 SQL의 기능을 직접 C언어 프로그램 중에 삽입할 수 있는 툴로써 고도의 어플리케이션 프로그램을 효율적으로 작성할 수 있다.

C-ISAM은 인덱스가 필요한 화일을 작성하고 조작하기 위한 C언어 라이브러리로 대량의 데이터를 취급할 경우, 고속으로 안정된 데이터 조작을 실현하여 효율적으로 보존할 수 있다.

그리고 한글 informix-4GL은 응용프로그램을 신속, 용이, 신뢰성 있게 개발할 수 있는 non-procedural문을 지원하며, 특히 데이터베이스 응용프로그램을 위한 기능과 사용자의 편의를 위해 프로시듀얼 기능도 지닌 프로그래머 향상을 위한 도구이다.

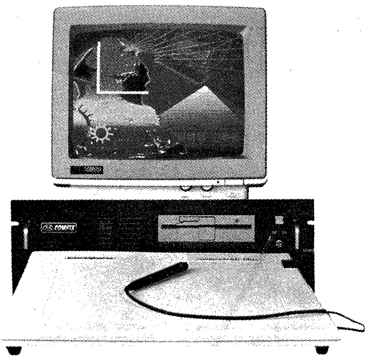
또한 동사에서는 한국 팔란티어(사장 : 김 춘)의 협조하에 386유닉스 머신인 Power-NET에 포팅 완료하여 멀티-유저용으로 릴리즈하게 되었다. 사용방법은 기존의 WP와 같아 새로 조작을 익힐 필요가 없고, 한자사서변환 및 완벽한 괘선처리가 특징이며, MS-DOS용 PWP와 완전한 호환성을 유지하므로 기존의 문서를 그대로 사용할 수 있다.

### -(주)컴픽스
## 「컴퓨터그래픽 문자발생기」 한국전자전서 대통령상 수상

제19회 한국전자전 신제품 경진대회에서 (주)컴픽스(대표 : 김광성)의 「컴퓨터그래픽·문자발생기」가 대통령상을 수상하였다.

이 시스템은 컴퓨터를 이용하여 그래픽처리를 효율적으로 관리할 수 있는 기기로서 서울올림픽 주방송사인 KBS의 중계방송시 TV화면에 그림·문자

●국내소식●



및 사진을 다양하게 매순간 전달, 시청감각을 높여 이미 인정받은 바 있다.

퍼스컴을 활용하여 컬러 그림을 원하는대로 그리거나 특정구조를 기억시켰다가 재활용, 프린팅할 수 있고, 한글·영문·한자·불어 등의 문자를 라인 및 preview 화면에 교대로 출력시키는 2페이지 방식을 채택하여 자유자재로 선택하여 사용할 수 있다.

특히 물감으로 표현하기 어려운 1천6백만 가지의 색상을 다양하게 표현할 수 있을 뿐만 아니라, 메뉴방식을 채택하여 키보드만 누르면 문자의 종류, 서체 등을 마음대로 선택할 수 있어 컴퓨터를 처음 대하는 사람도 손쉽게 활용할 수 있도록 설계하였다.

또한 종전에는 그래픽과 문자발생기가 분리됐던 것을 통합하여 개발하였으며, 특히 시스템의 핵심부분인 그래픽보드를 포함, 관련 소프트웨어를 외국기술에 전혀 의존하지 않고 자체 기술로 개발하여 하드웨어와 연결했다. 그리하여 가격면에서 외국제품에 비해 4배 정도 싼 2천만원대로 낮춤으로써, 외국제품과 가격경쟁력을 갖출 수 있게 되었다.

(주)컴픽스는 종업원수가 25명 정도인데 비해 연간 매출액은 15억 정도로 현재 국내에는 이 시스템이 90대 정도 보급되어 있는 실정이며, 앞으로 CATV 문자처리용 보드 및 관련 소프트웨어·다양한 서체 등을 개발하여, 해외시장 개척에도 적극 나설 계획이다.

그리고 90년 북경 아시안게임 주최국인 중국의 국영방송사와 자막처리 노하우 수출상담을 진행중인 것으로 알려졌다.

## -키컴
## 회계 전용 SW 2종류 개발

키컴(대표 : 김기복)은 회계전용 소프트와 재무관리통합 소프트웨어를 개발, 16비트 퍼스컴용 패키지로 상품화하여 판매를 시작하였다.

회계전용 소프트웨어인 「수퍼어카운트」는 일·월계표 합계잔액, 현금출납, 대차대조표, 손익계산서, 세금계산서 등 모든 회계업무를 전산처리한 시스템으로서 세무회계 사무소나 기업체, 경리부서 등에서 효과적으로 사용할 수 있다. 이미 개발되어 있는 5백여종의 다른 패키지와 비교하여 「수퍼어카운트」는 회계 기능을 다양화한 점이 특징이라 할 수 있다.

또한 「수퍼매니저」는 재무관리 통합 소프트웨어로서 중소기업의 회계관리를 비롯, 자재판매, 급여관리 등 전반적인 관리업무를 전산화한 시스템으로 자료입력이 간편하고 프로그램의 수정추가가 용이하다.

한편 키컴에서는 경영자문 및 의사결정까지 가능한 새 버전을 개발중이다.

## 삼보컴퓨터
## 퍼스컴 수출 1억불 달성

삼보컴퓨터(대표 : 김종길)는 국내 PC전문 생산업체로서는 최초로 연간 수출 1억불 실적을 달성하고 전문업체로서의 면모를 과시하였다.

지난 81년 퍼스컴 수출을 시작한 당사는 지난 10월말로 수출 1억4백만 달러를 기록, 올해 처음으로 연간수출 실적이 1억달러를 넘어 섰다.

한편, 동사는 늘어가는 주문량에 대비하여 반월 제1공장 근처에 9천평가량의 공장부지를 마련, 제1공장의 2배 규모의 제2공장 건설을 추진중에 있으며 제2공장이 완성되면 매달 15만대의 퍼스컴을 생산할 계획이다.

또한 시설확충과 아울러 소프트웨어 사업부를 신설하고 수출용 소프트웨어 개발과 해외전산화 용역사업에도 참여할 계획이다.

## -제우스컴퓨터(주)
## 김주동씨 상무이사로 승진발령

제우스컴퓨터(주)는 지난 10월 4일자로 영업담당 이사였던 김주동씨를 상무이사로 승진발령했다.

김주동 상무이사는 1946년 9월12일생으로 한양대 전기공학과를 졸업하고 한국전력과 ASEA(스웨덴회사인 전기종합메이커)에서 기술개발을 담당했던 전문엔지니어 출신이다.

## -한국오크스사
## 가정용 컴퓨터 개발

소프트웨어 개발 전문업체인 한국오크스(대표 : 김해일)사는 최근 일본 세가엔터프리세즈사와 기술 협력하에 8비트 가정용 컴퓨터인 세가마크III를 개발, 시판에 들어갔다.

이 시스템은 TV를 모니터로 사용할 수 있으며 주기억용량 40kB로, 음악소프트웨어를 이용, 작곡·편곡·연주가 가능하다.

한편 동사에서는 최근 가정에서 컴퓨터의 이용이 확대되고 있는 점을 감안하여 앞으로는 컴퓨터를 주식정보단말기로도 사용이 가능하도록 통신어댑터를 개발할 예정이다.

## 한국무역정보통신
## 카드형 모뎀 본격 수출

정보통신 역무제공업체인 한국무역통신(대표 : 주영훈)은 다이얼업모뎀다중화(MUX), 데이터송수신용 PAD 등 정보통신기기의 개발에 성공한데 이어, 미국의 로직모드사와 텔렉스용 모뎀카드 월1천대와 카드형 모뎀 월2천대씩을

장기 공급키로 계약, 시험선적을 끝내고 본격 수출에 들어 갔다.

수출하게 된 모뎀카드는 전이중 비동기 전송방식을 취하고 있으며 LED(발광다이오드)를 부착, 송수신 상태를 확인할 수 있는 것이 특징이다.

또한 동제품은 대미수출 확대를 위해 현재 FCC에 표준규격 승인을 요청중인 것으로 알려졌다.

한편 한국무역정보통신은 이번 대미수출을 계기로 정보통신기기 사업을 본격화하기 위해 공장부지를 물색하는 등 자체생산체계 구축을 서두르고 있으며 자체생산설비를 갖추는대로 형식승인을 거쳐 내수 판매에도 나설 계획이다.

### -미 가트너그룹
## 국내 대리점 설치

정보산업분야의 시장조사 전문업체인 미국의 가트너 그룹의 국내대리점으로 리서치아시아 코리아(대표 : 김영한)가 최근 설립되었다.

미국 가트너그룹은 지난 79년에 설립되어 세계각국의 컴퓨터 시장동향, 관련 기술변화, 중요기업체 정보 등을 제공하고 있는데 동 그룹은 컴퓨터 공급업체 뿐만 아니라 이용자측면에서 정보를 제공하고 있으며 특히 IBM 이용자를 위한 정보제공은 특이할 만 하다.

리서치아시아 코리아사는 가트너 그룹을 통해 세계 첨단기술변화와 해외시장 동향에 관한 정보를 국내업체에 회원제 형태로 공급할 계획이다.

아울러, 국내정보산업에 대한 시장조사보고서 발간업무와 시장조사에 대한 수탁 업무도 수행할 계획이다.

### -서울대 제어계측학과
## 제6회 전국 마이크로 로보트 경연대회 개최

지난 10월 29일 서울대 문화관 소극장에서는 서울대 제어계측공학과 주최로 제 6 회 전국 마이크로 로보트 경연대회가 열렸다.

로보트 제작동기를 부여하고 정보교환의 기회를 제공하기 위해 1983년 국내 최초로 대회가 개최된 이래 금년 6회째를 맞이한 동 대회에서는 총 34개팀이 참가를 신청하여 그 가운데 12개팀이 본선에 진출하였다.

경기결과 동 대회에서 입상한 로보트들은 예년대회와 비교하여 기술적으로 대단한 진보를 보였으며, 최우수상을 차지한 서울대 전윤호군의 SPEEDY3는 경연시간 16초 6 이라는 경이적인 기록을 수립하였다. 특히 우수상을 차지한 박남규군의 TURBO4는 스무드턴(smooth turn)이라는 곡선주행기술 개발에 성공하여 이 대회에서는 큰 박수갈채를 받았다. 또한 장려상에는 이준화군의 바람돌이, 김철균군의 HYPER 등이 차지하였다.

### 특허청
## 국내 우선권제도 도입

국내에서 개발된 첨단기술을 보호하기 위한 국내우선권 제도가 우리나라에서도 도입, 실시된다.

특허청에 의하면 우리나라 사람이 특허출원 후 1년안에 추가로 개발한 관련 개량기술을 당초의 출원내용과 함께 출원하는 경우 당초 출원에 포함된 부분은 처음 특허출원일로 소급하여 권리를 인정해 주는 국내우선제도를 도입, 시행키로 했다.

특허청은 우선권 인정에 관한 특허법 개정안을 내년중에 마련, 오는 90년부터 시행에 들어갈 방침이다.

국내우선권제도는 자국의 첨단기술 개발을 촉진, 보호해 주기 위한 제도로 현재 미·일 등 선진국을 중심으로 시행되고 있다.

### 전기통신연구소
## ISDN 관련과제, 기술전수키로

전기통신연구소(ETRI)는 ISDN 협대역 가입자접속 시스템등 최근 동연구소에서 개발한 4개 정부특정연구과제에 대해 기술 설명회를 개최하고 과제별 기술전수업체를 선정, 해당 기술을 전수키로 했다.

이번 ISDN 관련 기술전수에는 삼성반도체통신·금성반도체·대우통신·대한전선 등 대기업을 비롯 삼보컴퓨터·성미전자·성화전자·동욱전자·대영전자 등 18개 통신, 반도체, 컴퓨터 관련업체들이 대거 참여, 치열한 경합을 벌일 것으로 예상되고 있다.

이번에 전수될 과제별 내역을 보면, •ISDN협대역 가입자접속 시스템 기술의 경우 2B+D정보전송장치인 NTE (Network Termination Equipment) 제조기술을 비롯 터미널어댑터 및 터미널스위치 접속기술 •가입자 다중접선장치 분야에선 소·중규모의 ISDN접선장치인 멀티플렉서 제조기술 •PSTN(전화교환망)-PSDN(패킷교환망) 연동장치 기술전수 내역은 전화망·패킷교환망 접속제어장치와 연동유지보수 등 연동장치 개발이며 •통신용 장파장 발광소자 기술전수도 5백65M급 광통신 시스템 구축에 쓰이는 시제품의 설계·제작이 포함돼 있다.

### -효성컴퓨터
## 청담빌딩으로 옮겨

동양나일론(주)의 효성컴퓨터 사업부 TSC(Technical Support Center : 전시장, 교육장, A/S센터)는 지난 11월 5일 강남구 청담동 청담빌딩내로 이전했다.

효성그룹의 종합전산센터 운영의 일환으로서 그룹내 컴퓨터관련 지원업체가 모두 한자리에 모여 유기적인 협조체계를 구축함을 목표로, 안양 소재의 효성컴퓨터 사업부(전자영업부, 전자영업지원부, 전자수출부)도 12월초 동건물로 이전할 예정이다. PC

# 서클동정

국민대학

## 제6회 프로그램 전시회 어려운 환경속에서 무사히 마쳐

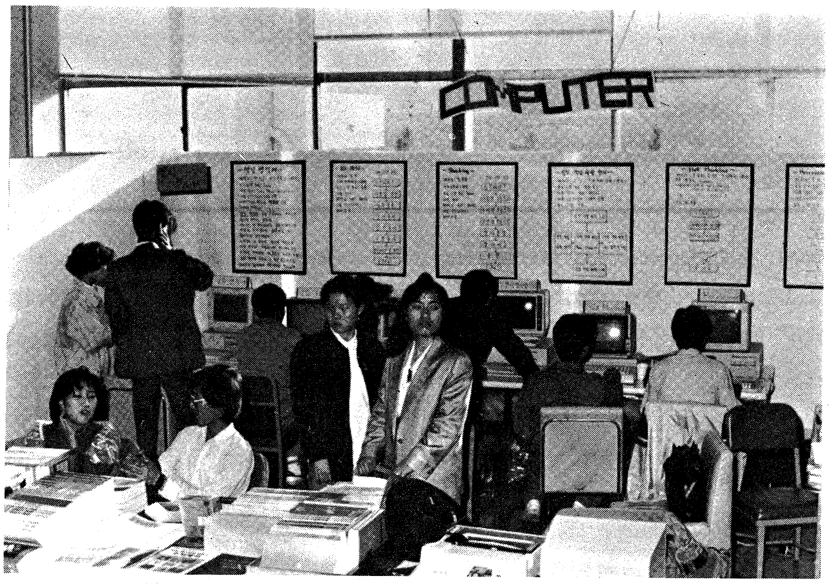
지난 10월 24일부터 26일까지 국민대학교 2호관 로비에서 국민대학교 컴퓨터 써클(K.C.C)의 제6회 프로그램 전시회가 개최되었다.

학사일정이 변경되어 전시회 일정에 차질이 생기고 기자재 구입에 어려움을 겪는 가운데 자신들의 컴퓨터를 제공하는 등 회원들의 단합된 의지로 열매 맺게된 이번 전시회에서는 터보파스칼 버전 4.0에서 구현한 게임 프로그램인 '옛날 옛적에', 애플소프트 베이직을 사용한 'I.Q Test', 기억력, 추리력, 상상력에 관한 게임 프로그램과 GW베이직을 사용하여 음악의 박자를 계산해 주는 '메트로놈'과 회사의 매일 업무에 관한 상태를 파악해 주는 '일일작업 현황관리' 등의 프로그램이 선보였다.

지난 '83년 '컴퓨터인의 저변확대', '컴퓨터 보급'이란 목적으로 창립되어 5월의 아트 전시회와 10월의 프로그램 전시회 등 매년 2회의 전시회를 개최해온 KCC는 학기중과 방학을 이용하여 베이직, 포트란, 코볼, 파스칼 등의 랭귀지에 대한 교육을 실시하여 실력향상에 주력하고 있다.

송창윤 회장(전자 2)은 앞으로 써클내에 더 많은 소프트웨어를 보유하도록 하고, 하드웨어 개발에 중점을 두며 기자재를 구입하기 위해 노력할 것이라고 한다. 또한 회원을 모집하는데 질적인 향상을 위해 교육적인 차원에서 모집하며 전 회원을 유저화하기 위해 교육을 강화할 계획이라 한다.

회원들이 자신의 컴퓨터 앞에 앉아 진지하게 프로그램을 설명하고 있다.



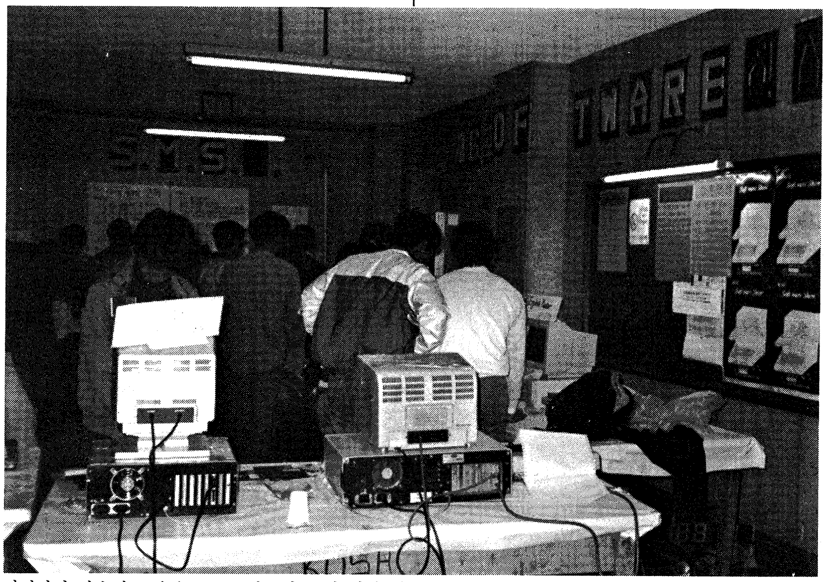

성균관대학

## 제6회 소프트웨어 전시회 성황리 마쳐

컴퓨터를 통한 첨단 과학기술과의 접촉과 창조적 능력의 향상을 목적으로 창단되어 6년의 연륜을 쌓아온 성균관대학교 마이컴 연구회(S.M.S.A)는 지난 10월 31일부터 11월 2일까지 교내 수선관 1층에서 소프트웨어 전시회를 가졌다.

동 소프트웨어 전시회에서는 주로 16비트(IBM XT/AT 호환기종) 중심의 소프트웨어가 선보였는데 이들 소프트웨어에는 MS-COBOL과 RM-COBOL을 사용하여 코볼에 대한 인식을 높이고자 ISAM 화일을 이용한 회원관리, 파스칼을 사용하여 소트의 종류와 방법을 알려주는 'Learn Sort', 디베이스 Ⅲ+를 사용하여 개인의 주소록이나 도서목록을 작성하고 이것을 소트하고 검색할 수 있는 기능을 가진 프로그램을 작성한 '개인 사무관리'와 써클 회원의 신상과 회비관리를 가능케한 '회원관리', GW베이직을 이용하여 영어 단어학습에 도움이 되도록 한 '워드 퍼즐'과 '영어 마스터', 닥터 할로우 II를 이용한 그래픽 엽서 등이 선보였다.

SMSA는 이와 같은 전시회 및 자체 실력을 향상시키기 위해 학기중에도 1주일에 2~3회의 베이직, 포트란, 파스칼, C 등의 언어에 관한 공부를 실시하



영어단어 학습에 도움을 주는 '워드퍼즐'이 가장 많은 인기를 끌었다.

고 있고, 방학중에는 어셈블러, 인공지능에 관한 세미나를 갖고 있다.

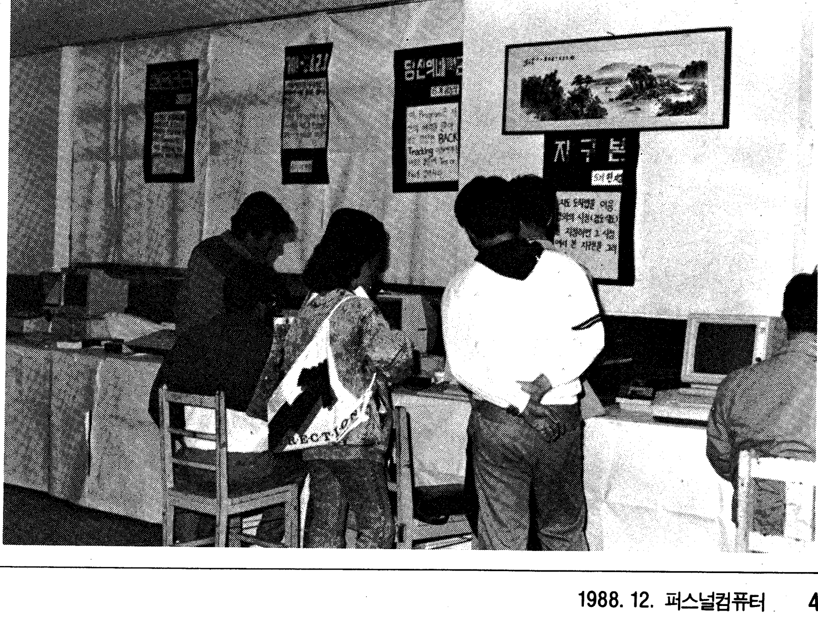
김대건 회장(금속 2)은 명륜 캠퍼스의 컴퓨터 인식을 높이고자 11월부터 매주 토요일 1시에 경상대에서 컴퓨터에 관한 기초 강좌를 실시할 것이며 전교의 컴퓨터 마인드 확산이라는 목적으로 자연과학 캠퍼스에만 있던 써클을 명륜 캠퍼스에도 설치할 예정이라고 한다. 또한 방학을 이용하여 초창기에 비해 약해진 하드웨어개발 부분에 중점을 두고 세미나를 통한 구체적인 결실이 맺어지도록 교육부분을 강화할 것이라고 한다.

## 숭실대학
## 제5회 컴퓨터 전시회
## 다양한 작품 선뵈

숭실대학교 컴퓨터 써클(S.S.C.C)은 지난 10월 12일부터 14일까지 숭실대학교 인문관 3층에서 제5회 컴퓨터 전시회를 열었다.

'83년 27명의 회원으로 창단하여 현재 200명의 회원을 확보하면서 지역사회와 학교에 공헌한다는 목적아래 모든 사람들이 컴퓨터를 생활화하는 것을 목표로 삼고 활동해온 SSCC는 매주 정기적인 세미나와 방학을 이용한 회원 및 비회원들의 연수를 통하여 전산실력을 향상 시키면서 학기중의 세미나를 통한 실력 테스트 및 컴퓨터 마인드 확산을 위해 매년 자체 전시회를 가져왔다.

당신의 매력포인트는 어디일까요.




이번 전시회에서는 주가변동을 입력시켜 증권 투자를 과학적으로 할 수 있게 한 주가분석, DB Ⅲ를 이용하여 효율적으로 회원을 관리하게 해주는 회원관리, 우주의 어떤 위치에서 지구를 바라 보았을 때 지구의 모습을 나타내주는 그래픽 지구본, 신상명세를 입력하면 이사가기 좋은 방향을 알려주는 701—2424, 궁합 등의 소프트웨어와 애플 컴퓨터를 이용하여 무선으로 콘트롤하는 환상의 88특급, 하루의 일정에 따라 그때마다 시간을 알려주는 특이한 제목의 '위험한 작품', 인체의 저항을 이용하여 센서를 통해 남녀 사이의 친밀도를 측정해 주는 러브 테스터 V2.0, 센서를 이용하여 전시장에 출입하는 인원을 관리해 주는 휴먼 체커 등의 하드웨어들이 선보였다.

SSCC의 이한승 회장(전자 2)은 동전시회를 준비하면서 어려운 점도 있었지만 작품들 모두가 회원들의 자발적인 노력과 정성으로 이루어졌다는데 큰 자부심을 가지며 앞으로의 계획에는 정기총회를 통해 차기 임원을 선출하고, 회지를 발간하는 일과 자체 실력을 향상시키기 위한 동계연수가 있다고 한다.

●서클동정●

## 한양대학 제6회 전자 전시회 마이크로 마우스 가장 큰 인기

한양대학교 전자공학과는 제6회 전자 전시회를 교내 반도체 센터에서 개최했다.

전자산업이 발달하면서 그 기대에 부응하기 위한 실제적인 지식이 필요하게 되었고 현재의 이론중심 교육이 실제의 현장에서는 적응하기 어렵다는 문제점을 보충하고자 실험을 통한 이론적 지식의 전개와 실력향상이란 목적과 한양대 전자과 자체의 단결과 전통을 세우고자 지난 84년부터 동 전자전시회를 개최하였다.

이번 전시회에서는 적외선 센서를 사용하고 구동에 스테핑 모터를 사용한 마이크로 마우스, RS-232C의 논리값 0과 1을 각각 1.5kHz, 4.5kHz로 바꾼 FM변조, 시각 출력시 필요한 음성을 음소별로 나누어 컴퓨터에 저장하였다가 출력 요청이 있을 때 이들 음소를 조합하여 음성으로 시각을 알려주는 장치, 적외선 센서를 이용하여 물체의 통

●화제의 얼굴



## 다양한 분야의 OA를 이룩하는데 젊음이 지닌 최대한의 노력과 신념으로 이바지하고 싶습니다.

"컴퓨터 소프트웨어의 대부분이 젊은 층으로 인해 탄생하고 있으며, 저희 자신들은 PC 붐 세대의 사람들이라고 생각합니다. 그래서 사회가 원하는 다양한 소프트웨어를 창조하고 싶었고 컴퓨터 마인드의 확산과 더불어 사회 전반적으로 시행되고 있는 다양한 분야의 OA(사무자동화)를 이룩하는데 젊음이 지닌 최대한의 노력과 신념으로 이바지하고 싶습니다."라고 하며, 옹골찬 의지로 20대 초반의 앳띤 모습을 지닌 5명의 젊은이(안상훈 : 성대 산공 휴학, 최승욱 : 성대 전기 석사1, 이두열 : 성대 전자 석사2, 남태희 : 성대 전자 졸, 홍재선 : 성대 금속 3)들이 모여 태성전산이라는 소프트웨어 하우스를 시작하였다.

같은 학교의 써클(성균관대학교 컴퓨터써클) 선후배사이인 이들은 아직 전원이 학교를 졸업하지 않았지만 개인이나 2인이 모여 83년부터 개별적인 아르바이트 활동으로 컴퓨터 운전면허시험 자동 테스터기, 유선방송 자막기, 유선방송 고객 및 대리점 관리 프로그램, 성적관리, 음주류, 섬유, 화장품회사의 관리 프로그램, 실험계측 프로그램, 기타 여러 방면의 소프트웨어 개발에 참여하면서 실력을 다져옴과 동시에 자신들의 학자금 및 경제적인 문제를 해결하는 등 실제적인 경험을 쌓아 왔다.

그리고 자신들이 하고 있는 일을 좀더 확대하여 발전시켜 보자는 생각을 갖게 되었고, 그러한 생각들을 모아 구체적으로 실현시켜 태성전산을 이루게 되었다. '인화, 실력배양, 창조, 신용'이라는 사훈아래 창사된지 1개월 정도된 이들이지만 소프트웨어 개발에는 결코 아마추어가 아닌 전문적인 기동력과 생산력을 가지고 있다고 자신한다.

이들을 대표하고 있는 남태희씨(26)는 회사의 조직이 5명의 사장(?)이라고 지칭하는 특이한 구성을 지니고 있고, 시장개척을 위해 수요에 의한 주문 생산과 철저한 시장 조사를 통한 기획상품을 개발하여 회사를 운영해 나갈 것이며, 보다 저렴한 비용으로 중소기업이 OA화 하도록 지원하고 싶고 무엇보다도 신용을 위해 전력투구할 것이라고 한다.

또한, 중소기업의 OA시장을 확대하기 위해 회계, 재무, 인사관리를 종합한 토탈 관리시스템을 보강하여 다양한 규모의 중소기업이 OA화 하는데 적합하도록 하는 소프트웨어를 개발할 계획이라고 한다. 이들은 현재 교회관리, 운수업무관리 등을 개발하고 있는 중이다.

개성 발휘, 창조력 발전, 성취감이 있기 때문에 소프트웨어를 개발하는 일을 한다는 너무나 젊은, 그러기에 미래가 밝은 태성전산의 젊은이들에게 기대를 걸어 본다.

과하는 방향에 따라 가산 및 감산하는 카운터, Z-80 마이크로 컴퓨터를 이용하여 가로 24, 세로 8도트의 칼라 매트릭스 리드, 음성을 아날로그-디지탈 콘버터로 양자화하여 메모리에 기억시키고 일정 시간이 지난 후에 디지탈-아날로그 콘버터로 음성을 재현하여 그때의 입력 음성과 가산하여 출력하는 디지탈 에코우, 20Hz~20,000Hz의 신호를 발생시켜 주파수 특성을 알고싶은 회로에 입력하여 그 출력을 분석하는 주파수 응답기 고급 앰프에서 채택하고 있는 OCL(Out Condenser Less)방식을 사용한 100W B급 앰프와, 반주문형(Semi-Custom) 반도체 설계방식 중의 하나인 게이트 어레이 방식을 위한 레이아웃 자동화 시스템이 선보였다.

한양대 전자공학과는 이러한 전시회를 준비하기 위해 학기중에는 이론교육과 방학중에는 제작교육을 실시하고 있으며 많은 사람들의 창작활동을 위하여 다양한 기자재들을 갖추고 있는 과실험실을 개방하고 있고 아이디어와 제작의욕을 가진 사람은 누구라도 환영하고 있다.

변상범 학회장(전자 3)은 이번 전시회의 모든 자료들을 기록, 보관하여 다음 전시회의 참고자료가 되도록 하여 더 발전할 수 있는 전시회가 되게 하겠다고 한다. 또한 겨울 방학을 이용하여 실무 교육에 중점을 두고, 화상처리 시스템에 관한 연구를 시도해 볼 계획이라 한다.

## 바이트 네트 PC통신 유저서클

현재 국내외적으로 일고 있는 PC통신 붐에 부응하여 순수한 아마추어 단체인 바이트 네트가 국내에서 처음으로 BBS의 보급을 유도하고 있다.

컴퓨터를 이용하는 사람들끼리 서로 의견을 교환하면서 다양한 정보를 얻을 수 있고, 공용의 소프트웨어를 이용하여 컴퓨터를 사용하면서 부딪치는 소프트웨어나 하드웨어상의 문제점을 해결할 수 있는 등 일종의 사용자간의 모임을 구축할 수 있다는 점에서 최근들어 급속히 확산되고 있는 PC통신 유저 서클인 바이트 네트는 지난 3월에 첫모임을 가지고 발족하였다.

현재 전국적으로 120여명에 이르고 있는 회원들은 의사, 교수, 학생, 회사원 등 다양한 직업과 연령층으로 이루어져 있고 참여도를 높인다는 의미에서 회원관리를 특별한 임원진이 하는 것이 아니라 전회원이 공동으로 관리하고 있다.

바이트 네트는 회원관리 및 통신접속시의 안정성을 위해 현재 미국에서 운용되고 있는 전문 호스트용 BBS 프로그램인 "Wildcat!"를 호스트용으로 운용하고 있으며, 메일교환 등에 필요한 한글 문제는 7→16비트, 16→7비트로의 자동교환 프로그램을 개발하여 전회원에게 배포하고 있고 이 두 프로그램은 공용의 소프트웨어로 회원관리를 원활하게 수행하고 있다.

회원제로 운영되고 있는 바이트 네트에 가입하려면 PC를 사용할 수 있는 사람이면 누구나 가능하며 모뎀은 필히 있어야 하며 회원관리 및 제반경비에 사용되는 소정의 월회비(1만원)을 납부하면 된다. 그리고 가입하게 되면 약 3000여개의 공용 소프트웨어를 지원받을 수 있다.

바이트 네트는 매월 1회의 정기적인 모임을 통하여 바이트 네트의 현황과 계획에 관하여 논의하고 있으며 내년 상반기까지 500명의 회원확보를 계획하고 있다. 회원가입을 원하는 사람은 (02)468—6838로 연락하면 된다.

## 디비 바이러스 데이터베이스광들만의 모임

모든 사람이 공유할 수 있는 데이터를 구축하여 업무 표준화를 이룩하기 위하여 구성된 모임이 있다. 이름하여 디비바이러스(회장: 박성훈 해외사업본부 수출1과).

88년 8월 사내 업무를 데이터베이스로 작성하는 사람과 개인적으로 데이터베이스에 관심이 있는 사람이 주축이 되어 첫 모임을 가진 이들은 전사원에게 데이터베이스를 감염시켜 사용하게 한다는 의미로 '디비바이러스'라는 이름을 가지고 있다.

디비바이러스는 세그룹으로 구성되어 있는데 1부는 업무를 분석하는 그룹 2부는 분석된 자료를 작성하고, 초보자에게 데이터베이스를 이해시키는 그룹 3부는 피교육 및 데이터베이스의 관련 자료를 수집하는 그룹으로 있으며 현재의 회원은 삼보컴퓨터의 사원으로 한정되어 있으며 데이터베이스에 관심있는 전사원에게 개방되어 있다.

이들은 현재 12명으로 구성되어 있고 정기적인 모임은 2주에 한번씩 가지는데 이때는 데이터베이스에 관한 전반적인 토론회 및 세미나를 하고 있고 1주일에 1회의 교육을 통하여 데이터베이스에 관한 기초강좌를 실시하고 있다.

현재는 회사의 신입사원 원서접수관리를 처리해 주고 있으며 실제의 업무프로그램을 작성하는 데이터베이스 프로그래머로서 보다 실무자로서 데이터베이스를 작성해 보려는 의욕을 지닌 회원이 많다고 한다.

연락처는 (02)3902—612이다.

## 엠팔 H-Mail 가입자 친목단체

'현대 첨단 문명의 이기인 컴퓨터가 인간들을 점차 소외시키고 있다'라는 통념에 도전장을 내걸고 현대 문명사회에서 표류하고 있는 현대인들을 서로 연결시키기 위한 모임이 있다. 이름하여 엠팔(전자 사서함 친구)!!!

지난 1988년 5월 데이콤의 H-Mail(한글 전자 사서함) 사용자 50인이 전자까페(홍익대앞 소재)에서 엠팔을 탄생시켰다. 엠팔은 남들이 보내는 메시

지를 자신의 이름으로 등록된 상자에 배달되도록 하는 사서함의 성격을 띠면서 완성된 메시지를 아주 빠르게 상대방에게 보낼 수 있고 동시에 여러 사용자에게 메시지를 전달하는 매스 미디어의 역할이 가능하며, 시간을 두고 전달할 수 있고 도착된 메시지를 프린터로 찍어내거나 혹은 디스크로 보관 하는 등 인쇄미디어가 가진 보관성을 그 특징으로 갖고 있는 H-Mail을 통해 대화하는 사람들이다.

엠팔의 통신방식은 회원의 집에 있는 모뎀을 통하여 전화선에 연결된 컴퓨터로 데이콤에 전화를 걸고 나서 회원의 ID(등록명:일종의 사서함 번호)번호를 컴퓨터 키보드로 누르면 회원의 컴퓨터와 데이콤의 컴퓨터가 연결되어 멀리 있어도 데이콤까지 찾아갈 필요없이 집에서 데이콤의 컴퓨터에 설치된 자신의 사서함을 볼 수 있게 되어 있다.

엠팔은 보다 나은 통신을 위하여 PC가 대형 컴퓨터의 인텔리전트 터미널의 역할을 하도록 하는 에뮬레이터를 개발하였다. 이것은 과거에 각 컴퓨터회사마다 독자적인 에뮬레이터가 있었지만 그 성능이 약하고 값이 비싼 것에 착안, 엠팔의 힘으로 개발하고 공용의 소프트웨어(Public Domain)로 개방하였다.

이들 에뮬레이터에는 묵현상 회원이 개발한 PC간의 통신 및 H-Mail 이용에 편리한 기능이 있는 'Revolt of E-mpals'과 장석원 회원이 개발한 PC간의 세팅기능을 위주로 만든 에뮬레이터로 데이콤의 천리안서비스를 받을 수 있게한 '파발마', 전영욱 회원이 개발한 일반적인 통신용 소프트웨어들이 할 수 있는 모든 기능을 하고 있는 소스가 T-Modem인 '메아리(Echo)', 최상순 회원이 개발한 8비트 애플용 통신 프

●미니 인터뷰/박순백/엠팔회장●



# 컴퓨터의 대중화에 앞장설 것입니다

"저희들은 온라인으로 항상 서로의 의견을 수시로 교환하고 있었으므로 그리 만날 필요를 느끼지 못하고 있었읍니다. 그러나 현재 H-Mail 서비스가 영상자료의 입출력까지는 지원하지 못하므로 한번 만나서 얼굴을 보고 싶다는 욕망들을 가지고 있었던 차에 전자까페가 홍익대 부근에 생기면서 그 업체를 경영하시는 분이 H-Mailer가 되심에 따라서 그곳에서 모임을 가지자는 의견이 온라인상에서 결집되어 1차 모임을 지난 5월에 갖게 되었읍니다. 그렇게 하여 모이게 된 엠팔의 산파 역할은 데이콤 전자신문 편집장이시고 저희 엠팔의 고문이신 유경희 데이콤 연구위원님이 하셨읍니다"라고 엠팔이 조직된 동기를 친근한 미소와 함께 설명한 엠팔의 회장을 맡고 있는 박순백 교수.

엠팔들과 대화하는 것이 생활의 한부분이 되었고 그것은 커다란 기쁨이라고 하며 지난 9월중 데이콤에서 아무런 사전통고 없이 바이너리(Binary) 화일을 전송하는 프로그램인 커밋(Ker-mit)을 호스트 컴퓨터에서 삭제한 일에 대하여 게시판 5번의 '전자회의' 게시판에 "개구리(여기서 개구리라고 한것은 커밋이 원래 The Muffet Show에 나오는 청개구리 이름이기 때문이다)를 살려다오!"라는 내용의 메일이 많이 게시되었고 곧이어 데이콤에서는 10월 15일자로 청개구리를 살려주었다는 재미있는 에피소드가 있었다고 한다. 또한 무엇보다도 잊을 수 없는 것은 엠팔회원들이 스스로 자신들의 에뮬레이터를 제작하여 이를 공용의 소프트웨어로 공개한 일로 이일이 매우 자랑스럽다고 한다.

엠팔은 현재 Ad hoc로 필요에 따라 모이지만 이를 기술적으로 지원하는 Empal's Choice Gr-oup은 수시로 모여 엠팔의 편의를 위해 여러 작업을 구상하고 있고 회원중에 50명 정도가 활발한 활동을 하고 있는데 앞으로 계속해서 회원을 늘려갈 생각이라고 한다.

"앞으로의 계획으로 BBS를 계획하고 있읍니다. 그리고 통신에 필요한 여러가지 프로그램을 발표하고 있는데 앞으로도 이런 작업을 계속하여 컴퓨터의 대중화에도 앞장설 것이며 특히 경험자의 집단으로서 Hard/So-ft Reviewer's Group으로서의 역할을 수행해 나갈 것입니다" 라고 박순백 회장이 말하는 엠팔의 계획과 이들에 거는 기대가 순조롭게 진행될 수 있으려면 전 엠팔들의 자체적인 충분한 능력을 구비하려는 노력이 계속되어야 할 것이며 누가뭐래도 문명의 이기를 접하는 선각자들인 만큼 이들의 의지는 시간이 지나면 지날수록 굳어갈 것이다.

로그램인 '워드콤' 등 4가지가 있다.

또한 앞으로는 BBS(Bulletin Board System)를 개설할 예정인데 이 시스템은 일종의 사설 통신망으로 현행법에는 허가되고 있지 않지만 앞으로 이 시스템과 관련된 통신역무업체의 출현이 예상되며 이 관계의 입법이 예고되고 있고 또 외국의 Nikkei Telecom(일경 텔레콤)이라든지, CompuServe 등이 내년(1989) 5월경에 진출할 것이 확실시 되는 시점에서 한국적인 BBS의 개설이 시급하다고 보아 준비중이다. 이름은 엠팔 BBS로 전자까페에 설치되어 운영될 예정이다. 이 BBS는 소스(Public Domain)를 수정하여 한글로 시험적으로 운영할 예정이며, 현재 상업적으로 팔리고 있는 C 랭귀지의 소스를 구입하여 이를 수정하려는 계획이다.

엠팔의 현재 회원은 전국적으로 약 200명 정도로 이루어져 있으며 엠팔이라는 용어는 현재 H-Mailer(한글 전자 사서함 사용자)라는 단어와 동일한 의미로 사용되고 있고, 정보문화의 확산이라는 차원에서 개방된 참가의식이 필요하다고 보고, 회원에 대한 자격제한을 그리 심하게 두지 않고 있으며 H-Mail 사용자중에 엠팔에 가입하고자 원하면 엠팔에 대한 정보를 얻을 수 있고 공식적인 모임에 참가할 수 있다.

엠팔회원의 연령층은 국민학생으로부터 노인층에 이르기까지 꽤 두터운 층을 형성하고 있으며 회사원, 학생을 포함하여 직업도 다양하다.

엠팔의 임원진은 고문에 유경희 데이콤 연구위원, 회장에 박순백 경희대 교수, 부회장에 이기성 장왕교재 연구원 상무, 간사에 한국 중공업의 강기웅 차장, 경리에 전자까페의 고연희씨로 구성되어 있다.

엠팔의 모임은 보통 H-Mail의 게시판을 통하여 공고되며 참석자의 범위가 정해지고 모임에서는 지난 모임 이후의 중요사안에 대하여 발표하고, 초이스 그룹에서 새로이 기획한 일을 발표하기도 하며, 회원들이 필요로 하는 프로그램을 복사하여 주기도 한다. 그러나 엠팔 회원간의 친목도모가 가장 중요한 부분을 차지한다. 그리고 PC통신을 통하여 필요에 따라 이합집산할 수 있는 충분한 능력이 있기 때문에 필연적(?)으로 정기적인 모임이 없다고 한다.

엠팔은 정보화시대를 살아가는 현대인에게 컴퓨터를 이용하여 정보를 교환할 뿐만 아니라 생활속의 이야기나 문학, 예술에 관련된 대화를 통하여 현대 기계문명의 정서적 빈곤을 해결하기 위하여 시작되었고, 비록 짧은 연륜을 가지고 있다고 해도 정보화 사회의 도래에 앞서서 많은 사람이 정보화 사회에 대한 준비에 만전을 기하게 하고 컴퓨터 마인드를 높이는 일에 주력할 것으로 기대된다.

엠팔에 대해 알고 싶은 사람은 전자까페(TEL 336—4585) 고연희씨에게로 연락하면 자세히 알 수 있다. PC

# 국외소식

## IBM
## RISC기술에도 특허보호 시작

IBM이 RISC기술에서도 특허보호의 자세를 취하기 시작하였다. Sun-4, Domain10000시리즈에 RISC기술을 각각 이용하고 있는 선과 아폴로 양사는 IBM으로부터 특허사용료의 지불을 요구하는 경고장을 받았다.

RISC기술은 70년대 중반에 IBM의 연구진들에 의해서 최초로 개발되었으며, 마이크로프로세서의 오퍼레이션을 간략화하여 격단의 성능향상을 가져온 기술로서 주목되고 있다. IBM은 RISC 관계로 병렬명령 실행기술 등 수10건의 특허를 취득하고 있으며, 기타 100건 이상이 신청중이다.

RISC의 로얄티 요구는 IBM이 금년부터 추진하고 있는 특허보호 정책의 일환이다. 또한, PC, PS/2 기술에서도 지난 4월에 호환기 메이커 각사에 로얄티 지불을 요구하였는데, 이 경우는 매상의 최대 5% 로얄티를 요구하고 있고 RISC도 이것에 준한다고 알려지고 있다.

타 분야에서도 점차 같은 정책이 적용될 것이 예상되며 IBM의 갑작스런 특허정책의 강화는 경합 각사에 위협을 줄 것으로 보인다.

## IBM
## PS/2용 RPG II 컴파일러 발표

IBM에서는 PS/2를 멀티유저 시스템으로 만들 수 있고, IBM System/36에서 PS/2까지 폭넓게 사용할 수 있는 미래의 소프트웨어를 발표하였다.

PS/2를 위한 RPG-II 컴파일러는 곧 발표할 대형 IBM의 중요한 한 부분으로서 기대되고 있다.

IBM은 마이크로 소프트사의 MS OS/2를 위한 Inc.사(CSPI)의 Baby/36 제품 버전인 캘리포니아 소프트웨어 프로덕트를 곧 제공할 예정이다. CSPI사의 대변인 Mark Toennission은 IBM과 CSPI 모두 OS/2 관련 시스템을 운영하기 위해 개발된 패키지들을 가까운 시일내에 발표할 것이라고 말하였고, 또한 "IBM은 PS/2와 함께 미니컴퓨터를 개발하려고 시도하고 있다"라고 덧붙였다.

IBM 대변인은 IBM사가 이미 PS/2에 RPG-II를 응용하기 위한 설비를 제공할 의사가 있다는 뜻을 비춰왔다는 것을 확인하면서 "IBM은 곧 이 사실을 정확히 발표할 예정이다"라고 단언하였다.

IBM 제품은 PS/2 RPG-II 어플리케이션 플래트폼이라 부르며, CSPI버전은 OS/2를 위한 Baby/36이라 부르게 될 것으로 알려졌다. IBM 버전은 단지 PS/2에서만 수행이 가능한 반면에, CSPI 버전은 IBM PC 호환기종에서 뿐만 아니라 PS/2에서도 수행이 가능하다.

## 유니시스사 등 8개사
## OSI 표준화 지원조직 설립

구미의 통신업자와 컴퓨터 메이커 8개사는 OSI(개방형 시스템간 상호접속)의 네트워크 관리기능을 개발하는 새로운 조직 「OSI/네트워크 메니지먼트 광장」—OSI/NM 포럼—을 결성하였다.

설립 멤버는 AT&T, 아무딘, 브리티슈 텔레콤, 휴렛팩커드, 유니시스 등 8개사로 구성 되었다.

동 포럼은 ISO로 정식 네트워크 관리기능의 표준이 나와있으므로 과도적인 관리기능의 표준을 작성하는 것을 목적으로 한다. 18개월내에 표준규격을 작성하여 데몬 시스템을 가동, 90년경에 각 메이커로부터 동표준에 따른 OSI 대응제품이 등장하도록 할 것이라 한다.

표준화는 두가지 측면에서 추진되는데, 첫째는 관리정보를 수집하기 위한 프로토콜의 표준화로, OSI의 하위 3계층에 대해서 패킷망의 표준인 X.25와 이더네트 LAN의 IEEE 802.3을 채용한다. 또, 상위층(4~7층)은 ISO로 규격초안이 되어 있는 CMIS/P(공통관리정보/프로토콜)을 기초로 개발한다.

두번째는 수집의 대상인 관리정보의 내용구성을 파악하는 것이다.

일본 통산성
## 이기종간 호환성을 실현할 슈퍼 시그마계획 추진

일본에서는 내년부터 5개년계획으로 기종이 서로 다른 컴퓨터간의 호환성실현, 차세대 컴퓨터 프로그램 개발 등을 위한「슈퍼 시그마계획」이 추진된다.

점차 심각해지는 소프트웨어기술자 부족현상을 크게 해소하는데도 기여할 이 계획은 소프트웨어의 생산성 향상에 역점을 둔 5년간의 시그마프로젝트가 내년으로 끝남에 따라 이의 연속사업으로 진행된다.

계획내용을 살펴보면 호환성에 관해서는 5년후를 목표로 이기종 컴퓨터간에 소프트웨어를 주고 받을 수 있는 접속장치 구축, 호환기종이 되는 기본소프트웨어(OS)의 방식결정 등이다. 또한 차세대 언어 개발에서는 시그마프로젝트에서 구축한 시스템을 배경으로 일반인들도 프로그램언어 사용이 가능하도록 한다는 것이다.

일본 통산성은 이 계획이 소프트웨어 개발의 종목투자 방지, 개발기간 단축 등의 효과도 가져올 것으로 보고 있다.

뉴욕주
## 미국에서 처음으로 VDT 규제법 제정

금년 6월 뉴욕주에서는 미국에서는 처음으로 VDT(비디오 디스플레이 터미널)규제법이 제정되었다. 입법당국에서는 VDT가 이용자가 건강을 해친다고 하여 이러한 규제를 정하였는데 이에 대해 각 메이커에서는 지나친 과장이라고 반발하고 나섰다.

법안의 내용을 보면, 고용자측에서는 VDT로 인해서 생기는 종업원의 장해에 대해서 치료비의 일부를 지불하도록 명령하였으며, 또한 그에 따른 노동환경의 개선도 의무화하고 있다. 과거 VDT장해를 제거하기 위한 여러가지 조사가 행하여졌으나 결론을 맺지 못한 상태에서 서포크군의 입법안이 제시된 것이다.

VDT의 유저와 메이커 쌍방의 연합조직인 워싱턴의 오피스 테크놀리지 센터(Office Technology Center)의 이사장은 "VDT기기가 안전하지 않다고 선언한 것과 같다. 그러나 지금까지의 조사에서는 모두, VDT는 어떠한 장해도 야기되지 않는다는 결과가 나왔으며, 지난 10년동안 미국의 26주에서 각각의 규제 법률안이 검토되었으나 어떠한 것도 실효를 거두지 못하였다"고 반대의 입장을 밝혔다.

또한 IBM에서도 금년초 개최된 동법의 공청회에서 반대 의견을 표명하고 나섰다. IBM사의 스폭스우먼씨는 "법률은 불필요하다. VDT이용에 의한 건강장해는 없으며, 이 문제는 쾌적한 환경에 관계되는 것으로 법률에서 이렇다할 문제는 아니다"라고 서술하였다.

한편, 다른 많은 메이커에서도 "VDT의 논의는 이제까지 노사간의 화합으로 원만히 해결할 수 있는 일"이라고 밝혔다.

EC 등 선진각국
## 특허출원 규격 통일키로

지난 11월1일 일본 동경에서는 선진국의 특허제도 통일을 협의하기 위한 회담이 열렸다.

본 회의에서는 미국·EC·일본의 특허청장들이 모여 특허권자 출원규격 통일에 관해 협의, 다음과 같이 합의하였다.

• 컴퓨터를 사용해 특허 심사하는 전자출원의 규격을 미국·유럽·일본이 통일한다. • 반도체의 구조와 제조방법 등 관련된 복수의 특허를 1회 출원으로 마치도록 미국·유럽·일본의 제도를 조정한다는 등이다.

한편 이 회담에 참석한 쿠이그 미국특허상표청장과 브렌들리 EC특허청장은 일본 통산성장관과 회담하면서, 현재 9백명인 일본 심사관을 증원시켜 3년이상 걸리는 특허심사기준을 단축해 줄 것을 요청하였다.

또한 생명공학등 첨단기술 특허의 취급에 대해서는 미국·유럽·일본이 각각의 방침을 정리한 후 내년초에 다시 회합을 갖기로 했다.

영국 ROCC 컴퓨터사
## 워크스테이션 관리시스템 개발

한대의 컴퓨터가 독립형 주컴퓨터 및 응용프로세서로서 뿐만 아니라 대형컴퓨터와 접속된 네트워크를 구성하여 화일정보 제공장치 및 통신제어용 컴퓨터로서도 활용될 수 있도록 한 워크스테이션 관리시스템이 개발되었다.

이 WMS/X시스템은 영국의 ROCC 컴퓨터사에 의해서 개발된 것으로 개방시스템 접속(OSI)의 표준방식과 UNIX 오퍼레이팅 시스템을 채택하고, 모토롤라68000 마이크로프로세서와 RISC 칩 기술을 활용하고 있다. 이 시스템은 데이터, 문자, 그래픽을 다루는 정보처리를 효과적으로 지원할 수 있도록 설계되었으며, 8명부터 32명까지의 사용자가 동시에 이용할 수 있는 세계 최고속의 성능을 지닌 UNIX 컴퓨터시스템으로 평가받고 있으며, 일반적인 사무용의 업무용도에 거의 모두 대응할 수 있는 포괄적인 기능의 소프트웨어를 내장하고 있다.

또, ROCC사가 개발해 낸 4세대의 프로그램언어인 C-CheckII와 관계형 데이터베이스, UNIX 오퍼레이팅 시스

템에 대한 전체적인 시스템접속 관리기능을 제공해 주며 완전한 네트워크구성 기능도 지녀 이더네트 및 X25접속과의 호환성도 가지고 있다.

## 3Com사
# 새로운 네트워크 시스템 발표

3Com사에서는 자사의 3+네트워크 운영시스템의 현재 유저들을 마이크로소프트사의 OS/2 LAN Manager에 기반을 둔 3+Open 차세대 소프트웨어로 이식시키기 위한 전략을 발표하였다.

초기의 버전은 무제한의 유저들을 위한 3+Open LAN Manage AS(Advanced System)와 5명 정도의 유저들을 지원할 수 있는 엔트리 시스템 등을 포함하고 있다. 반면에 이식 프로그램은, 3+Open AS 유저들을 대상으로 하며 네트워크 서버가 PC를 적용할 것인지 3S/400, 3S/200 및 3서버3과 같은 3Com사의 서버를 적용할 것인지에 따라서 다르게 설계되어 있다.

OS/2는 MS-DOS를 지원하고, 3Com사는 그의 서버상에 표준 DOS디스크 포맷을 사용하기 때문에 소프트웨어를 변경할 필요가 없다.

이러한 이식 패키지는 PC 서버를 위해서 3+Open LAN Manage AS를 제공하며, 이것은 서버를 위한 OS/2 소프트웨어와 DOS 및 OS/2 워크스테이션을 위한 소프트웨어를 포함하고 있다.

## 미, 이노베이션 인터내셔널사
# 소련과 합작으로 컴퓨터회사 설립

미국의 투자은행 이노베이션 인터내셔널사의 자회사인 이노베이션 컴퓨터사는 지난 10월 19일 소련 무선공업성과 퍼스널컴퓨터를 생산, 판매하는 합자회사를 설립키로 합의, 의정서에 조인하였다.

합의된 의정서의 내용을 보면, 소련국내에서 판매하는 퍼스컴의 수입, 소련국내에서의 퍼스컴 조립 및 제조, 소련·서유럽 각국에서의 퍼스컴 판매 등이다.

그러나, 이러한 내용들은 대공산권수출규제 위원회(COCOM)규정과 관련, 미정부의 허가를 받아야 한다.

한편 이날 이노베이션 인터내셔널사의 프랑크·리이트 사장은 자회사의 설립의의를 「미·소상업 교류의 극적변화가 세계의 주목을 받고 있는 가운데 이번 합작사업은 역사적 의의를 갖는다」고 밝혔다.

## 영, 텔레콤사
# 세계 최대규모의 OA프로젝트 계획

분산형 사무자동화 시스템 구축 계획으로는 세계 최대 규모인 OA프로젝트가 최근 British Telecom사에 의해 착수되었다. 이 회사는 90년대의 기간동안 이 계획의 추진을 위해 약 2,500만~3,000만파운드의 자금을 매년 투입할 계획이다.

COAST(Common Office Automation System for Telecom)로 불리는 이 사무자동화 시스템이 구축되면 시스템의 사용자는 전세계적으로 약 6만명 이상에 달할 것이며, 관리효율의 대폭적인 개선과 구매경비의 감소로 원가절감이 이뤄질 것으로 기대되고 있다.

이 시스템은 British Telecom사가 자회사인 Fulcrum Communication사를 통해 독자적으로 개발해 낸 슈퍼 마이크로 컴퓨터를 활용하여 이를 기존의 다양한 소형 컴퓨터 및 지역 네트워크와 접속시키며, 궁극적으로는 이들 소형 컴퓨터 및 지역 네트워크를 대체할 계획이다.

이 시스템은 또한 각 지역의 컴퓨터 설비와 결합되어 전자우편, 워드프로세서 기능, 일정계획 등의 서비스를 곧바로 제공해 줄 것이며, 사용자는 이를 통해 데이터 및 텍스트, 그래픽 정보의 모든 종류를 신속하고 효율적으로 처리, 교환할 수 있게 된다.

## 영, 사이언스시스템사
# 우주선제어 패키지 개발

태양 및 목성의 탐사를 위해 유럽우주기구(ESA)가 발사할 계획인 율리시즈(Ulysses) 우주선의 미세한 움직임을 제어하기 위해, 영국에서 개발된 소프트웨어 변환패키지가 활용될 예정이다.

율리시즈 우주선은 ESA와 미국의 NASA가 공동으로 참여, 오는 90년 10월 5년 동안의 비행예정으로 태양과 목성 탐사를 목적으로 발사될 계획이며, 고도와 궤도를 제어하는 비행동력 소프트웨어는 우주선이 정확한 궤도에 진입함과 거의 동시에 기능을 발휘하게 된다.

다른 복잡한 움직임은 목성의 남극지점 밑에서 우주선이 태양계의 정확한 궤도로 옮겨가기 위해 중력의 도움으로 이뤄지는 분사장치를 이용한 이동으로, 이같은 작동도 소프트웨어에 의해 정밀 제어된다.

우주선의 주컴퓨터를 작동시키는 원래의 비행동력제어 소프트웨어는 우주선이 업무를 마치고 소멸될 때까지 유

지관리가 전혀 불가능하므로 기본 소프트웨어를 변환시키는 것이 이 패키지의 특징으로, 서부잉글랜드 브리스톨에 소재한 사이언스 시스템사가 개발했다.

변환된 소프트웨어는 VAX컴퓨터에 내장되어 우주선이 급격한 움직임을 계속하는 동안 고도와 궤도를 올바르게 유지하도록 제어하는 역할을 하게 된다. 연구팀은 우주선 발사에 앞서 NASA의 제트추진연구소와 공동으로 컴퓨터시스템에 소프트웨어를 집적시키는 시험을 추진할 계획이며, 최초 운용기간 동안 제트추진연구소가 기술지원을 실시하게 된다.

## AT&T사
# 선과의 협약 파기

최근, AT&T사는 앞으로 생산될 컴퓨터(유닉스 시스템V)에 선 마이크로시스템사의 SPAR(Scalable Processor Architecture)칩을 사용하기로 한 협약을 철회하고, 모토롤라사의 88000 RISC-BASED 마이크로프로세서를 채택하기로 했다.

선과 AT&T의 관계는 지난 1월 체결한 협약에 따라 시작되었는데 이 협약은 컴퓨터 업계에 커다란 파문을 가져왔다. AT&T내부에서는 마이크로프로세서의 주 공급회사인 모토롤라사와의 오랜 관계가 위협을 받을 것이라는 우려가 생겼고, 외부적으로는 아폴로, BULL, DEC, HP, IBM, NIXDORF, SIEMENS 등 7개의 유수업체들이 OSF(Open Software Foundation)을 창설하여 선과 AT&T에 대항하게 되었다.

이에 AT&T의 새 공동회장으로 취임한 ROBERT ALLEN과 KAVER는 선과의 동맹에 대해 재검토를 하게 되었고 결국 협약을 파기하게 된 것이다.

특히 AT&T는 자사의 미니컴퓨터 3B라인에 모토롤라의 68000프로세서를 사용하고 있으며, 이 밖에도 몇개의 제품에 있어서 모토롤라와 장기적인 관계를 맺어 오고 있었다. 그러나 선과의 계약은 이렇게 장기간 신의를 지켜온 동맹업체 뿐만 아니라 그들과 연결된 핵심사업에 큰 어려움을 초래하게 되었으며, AT&T의 주된 이미지를 흐리게 하였다. 또한 AT&T는 자체 내에서 고성능 워크스테이션을 개발하고 있는데 만약 그 제품이 선 마이크로시스템의 제품과 경쟁하게 된다면 이것 또한 큰 문제가 될 것이다.

결국 AT&T로서는 선보다는 모토롤라나 인텔과 협력하는 것이 여러면에서 이득이 있을 것이다.

## 미, 익스카리버사
# 문자형 패턴 검색에 적합한 패키지 개발

미국의 익스카리버사는 패턴이나 문자열의 검색에 적합한 뉴럴네트 패키지 Savvy를 개발, 판매를 개시하였다.

이 패키지는 모델 APRP(Adaptive Pattern Recognition Processor)을 기초로 문자열 검색 등에 적합한 기능을 추가하여 동작하도록 하였는데, 특히 검색속도가 빠른 것이 특징으로 MicroVAX상에서 1200페이지(4.4MB) 정도의 문서에서 어떤 문자열을 검색하는데에 3초가 걸린다.

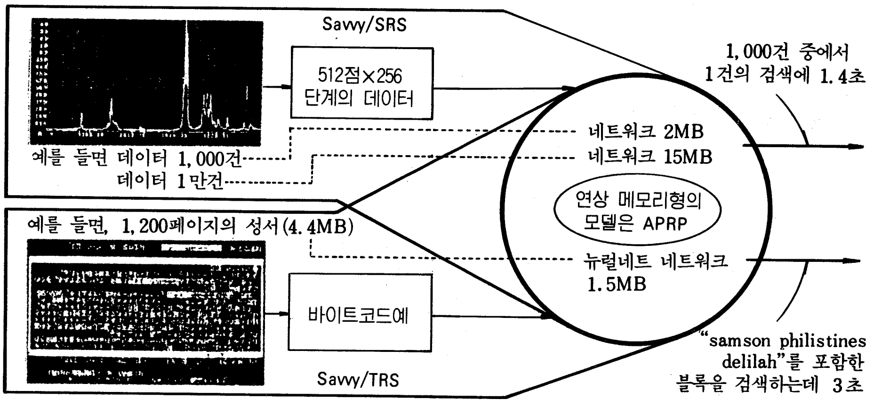
**그림** 1에서 보는 바와 같이 문자열의 검색용은 Savvy/TRS, 패턴 검색용을 Savvy/SRS라 하고 한편 영상인식용 Savvy/RIS도 시험, 제작이 끝나 IC의 식별에 이용될 것이라고 한다.

패턴 검색이란 그림 1과 같이 측정결과(그래프)를 예를 들어 1000건을 입력할 경우 거기에 새로운 그래프를 주어 비슷한 것을 취해내는 처리를 말한다. 문자형 패턴 검색에서는 미리 측정한 데이터를 기초로 품질의 순위를 부여하는 등의 응용이 활발히 이루어지고 있다.

패턴 검색용 SRS에는 횡축 512점, 종축 256치의 데이터(그래프에 대응)가 입력되며 1패턴의 데이터량은 1kB이다. 이와 같이 패턴 1000건을 입력할 경우 이것으로 부터 데이터 1건을 검색하는 시간은 MicroVAX상에서 1.4초가 걸릴 뿐이다.

문자열 검색의 TRS는 키워드검색 등으로 응용할 수 있는데, 예를 들어 1200페이지 상당의 성서를 1페이지 정도의 블록으로 나누어 뉴럴네트에 입력할 경우 블록의 검색은 3초만에 실행된다. 단, 검색시간은 검색하는 문자열에 의해 변한다.

〔그림 1〕 Savvy의 동작



NEC
## V33용의 부동소수점연산 코프로세서 개발

NEC는 16비트 마이크로프로세서 V33 전용의 부동소수점연산 코프로세서(FPU) μPD72291을 개발, 지난 10월부터 출하를 시작하였다.

V시리즈는 전용 FPU를 가지고 있는데 16비트의 V시리즈를 사용한 시스템에서 FPU가 필요한 경우는 미국 인텔사의 8087과 조합시키는 것이 일반적이다.

그러나 8087과 V시리즈에서는 동작클럭의 알고리즘이 다르므로 10MHz판의 8087은 V시리즈와 조합시키면 8MHz에서밖에 사용할 수 없다는 제약이 있었다.

72291은 삼각함수 8종류, 지수함수, 대수함수 등 숫자함수 13종류와 더불어 3요소와 4요소의 벡터내적 명령을 갖추고 있다(표 1 참조). 또 IEEE754에 의거한 단정도 및 배정도 데이터가 사용되며 확장배정도 연산은 지원하지 않는다. 이 때문에 64비트폭의 연산레지스터가 내장되어 있다.

단정도연산의 속도를 향상시키기 위하여 단정도데이터는 72291내부에서도 단정도데이터를 그대로 연산하고 있다. 16MHz에서 동작하는 72291은 단정도의 4칙연산을 1.8μs, 평방근연산을 2.4μs에서 실행한다(표 2 참조).

이 값은 8MHz에서 동작하는 8087을 V30과 조합하여 사용한 경우에, 가감산에서 11.3μs, 승산에서 13.8μs, 제산에서 26.9μs, 평방근연산에서 22.5μs를 요하는 것에 비교하면 훨씬 빠른 속도이다. 그리고 V33을 72291에 조합한 구성에서 종합성능은 V30+8087보다도 빠른 5~6배를 얻을 수 있다고 한다.

〔표 1〕 V33이 지원하는 숫자함수와 벡터명령

| | |
|---|---|
| 삼각함수 | sin θ, cos θ, tan θ<br>sin θ와 cos θ의 동시계산<br>arcsin θ, arccos θ, arctan θ<br>arctan θ(Y/X)와 $\sqrt{X^2+Y^2}$의 동시계산 |
| 지수함수 | $e^x$, $e^x-1$ |
| 대수함수 | ln X |
| 평방근 | $\sqrt{X}$ |
| 누승 | $X^y$ |
| 벡터내적 | 3요소, 4요소 |

〔표 2〕 V33의 연산 실행시간

| | *단정도연산 | 배정도연산 |
|---|---|---|
| 가감승산 | 1.8μs | 1.8μs |
| 제산 | 1.8μs | 2.4μs |
| sin θ | 5.7μs | 7.6μs |
| 평방근 | 2.4μs | 4.2μs |
| 4요소 벡터내적 | 1.8μs | 2.1μs |

후지쯔
## FACT Ⅲ시리즈 판매 개시

후지쯔에서는 89년 2월부터 일본에서 개시되는 주 5일 근무제에 대응해 무인화운용기능을 강화한 자동거래장치〔FACT Ⅲ시리즈〕의 판매를 개시하였다.

동 시스템은 무인운용기능외에 계원이나 이용자 취급방법의 간편화, 소형화에 의한 설치면적의 유효이용 등을 실현하였다.

72년부터 동사에서는 자동거래장치의 판매를 시작해 총 4만5천대 정도를 보급한 상태이며, 새로 판매될 FACT Ⅲ는 지난 80년에 판매 개시됐던 FACT Ⅱ의 후속 기종으로서 자동거래장치 시장에서의 최고점유율을 고수할 방침이다.

동 시스템은 무인운용기능 강화·전체구성 유니트의 대폭적인 소형화·장기간 사용에 대응한 CD 제공 등을 특징으로 들 수 있다.

미국 상무성
## 서방에 전산화시스템 지원

미국은 전략상품의 대공산권 유출을 막기 위해 영국·서독·일본 등 서방각국의 수출검사제도 컴퓨터화를 지원할 예정이다.

미국의 상무차관인 폴 프리덴버그씨에 의하면 "서방 각국은 그들의 시스템을 현재 미국이 채용하고 있는 시스템으로 개선하기를 원한다."고 밝혔다.

한편, 현재 미국이 채용한 컴퓨터시스템은 새로운 광학문자 판독장치로서, 물품이 신청될 경우 이를 자동적으로 판독, 메모리시켜주며 허가관리들은 허가여부를 컴퓨터터미널에 입력하게 된다.

또한 이 입력된 정보를 컴퓨터를 통해 파악함으로써 세관관리들은 서류없이 정확하게 수출품의 선적여부를 결정할 수 있다.

그리고 새로운 컴퓨터 시스템 채용으로 인하여 현재 4백건의 허가처리 수준을 1천2백건으로 끌어올릴 수 있을 것이라고 한다.

제가 금년중에 확립될 것이다

RISC CPU 시장규모는 1천6백만 달러(1987년)에서 4천만달러(1988년)로 불과 1년사이에 2.5배나 증가했고, 4년후인 1992년에는 7억달러(1987년 대비 44배 증가) 수준으로 확대될 것으로 예측된다.

아폴로컴퓨터

## 닉스돌프와 워크스테이션 판매계약 체결

미국의 아폴로컴퓨터사와 서독의 닉스돌프사는 최근 아폴로의 워크스테이션을 닉스돌프가 판매할 수 있도록 하는 워크스테이션 판매계약을 체결했다.

이로서 닉스돌프사는 서독을 비롯 세계 전지역에서 아폴로 혹은 자체상표를 부착해 공급할 수 있게 됐다.

아폴로와 닉스돌프는 OSF(오픈 소프트웨어단체) 창립업체로 컴퓨터업체들이 공급하는 하드웨어와 소프트웨어를 사용하기 편리하도록 오픈 소프트웨어를 개발, 제공하는데 공동 노력하고 있는데 이번의 판매계약도 이러한 차원에서 이루어진 것이다.

한편, 아폴로컴퓨터는 닉스돌프외에도 멘토그라픽스 지멘스 오토트롤 등에 아폴로 워크스테이션을 공급, 각사가 자체상표를 부착해 판매하도록 하고 있다. 이들 업체는 적용 분야별로 고유의 응용소프트웨어를 개발, 워크스테이션에 탑재해 공급하고 있다.

(주)도시바

## TRON사양에 의한 32비트 μP 개발

(주)도시바는 TRON 규격에 의거한 32비트 마이크로프로세서로서 복수의 명령에 관해서 동시에 연산처리를 행하는 「비동기 파이프라인 처리」나 1μ CMOS 미세가공기술 등에 의해 클럭주파수를 25MHz로 고속화하고, 평균 명령 실행속도 5MIPS의 성능을 실현한 「TX1」을 개발, 샘플제품을 만들었다.

이 샘플제품의 설계에는 1μ 룰의 2층 알루미늄배선 CMOS 프로세스를 채용하고 있고, 칩사이즈 10.89×10.27㎜사이에 45만개의 트랜지스터를 집적하고 있다.

명령세트는 TRON 칩의 기본규격 중에서 고속성을 중시하는 시스템제어 등에 적당한 「LIR」레벨을 완전하게 포함하고, 특히 TRON 칩과의 상호호환성을 유지하기 때문에 유효한 명령을 추가할 경우 92개까지 가능하다. 더구나 TRON규격에 추가하여 프로그램의 에러 발견에 편리한 디버그 전용 레지스터를 12개 구비하고 있다.

응용분야로서는 고속성을 살린 기계나 플랜트 등의 리얼타임 시스템제어용, 로보트제어용, ASIC의 코어프로세서용 등 차세대의 기기에 있어서 센트럴 프로세서로서의 이용이 고려되고 있다.

유니시스

## 분산형 통신 프로세서인 DCP/30 발표

미국 유니시스사는 최근 분산형 통신 프로세서(DCP)인 DCP/30이라는 새로운 중형 모델을 발표하였다.

동 기종은 매초 45개의 거래(Transaction)를 처리할 수 있는데, 그 처리속도는 DCPS기종의 하위모델인 DCP/15의 3배, 상위모델인 DCP/50의 3분의1에 달하며, 네트워크 기능 외에 프론트 엔드 및 nodal 프로세싱 기능을 갖고 있다.

또한 타 DCP 모델과도 완전한 호환성을 갖고 있어 고객의 통신시스템 성장에 따라 계속 확장시킬 수 있다.

DCP/30의 설계는 최신의 게이트어레이기술, IBM 메모리회로에 기초하고 있어 신뢰도나 유지·보수에 용이하고 아울러 운영비용을 절감할 수 있다.

동 시스템의 기능을 살펴보면 16개의 라인 모듈을 지원하는 I/O 모듈을 접속함으로써 11개의 기본 라인 모듈을 확장할 수 있고 분산형, 별형, 망형 네트워크를 지원하며 컨센트레이터, 프론트 엔드 프로세서 사이의 데이터 교환을 고속 처리한다.

SPARC

## 32비트 WS시장의 주역으로 등장

RISC 아키텍처 32비트 CPU시장이 모토롤라 68000, 인텔 80386의 범용 CISC 마이크로프로세서 시장을 곁눈질하며, 급속도로 확대되고 있다.

특히 시장에 등장한지 1년에 불과한 Sun Microsystems(이하 Sun) 주도의 SPARC의 진출이 주목된다. SPARC는 이미 반도체메이커 5개사에 의해 라이센스 생산되며, 10MIPS에서 50MIPS까지의 각종 고속 CPU 공급체

# 직접모드와 간접모드

이기성/장왕교재연구원 상무

## 베이직언어

컴퓨터 프로그래밍언어(이하 언어라 표기함)에는 코볼, 포트란, 파스칼, 어셈블리어, 베이직, 시(C) 등 여러가지가 있어서 컴퓨터를 처음 배우려는 사용자들이 종종 당황하게 되는 수가 있다. 게다가 간혹 주위에서 어느 언어가 좋고 어느 언어는 나쁘다는 소리도 흘러나와 더욱 언어선택을 힘들게 만들곤 한다. 그러나 이렇게 자기의 입장에서 언어를 평가하는 것은 올바른 시각이 아니다. 모든 언어들은 나름대로 독특한 특징들을 가지고 있어서 처리업무의 성격에 따라 제각기 그 진가를 발휘할 수 있다. 즉 수치계산에 적합한 언어가 있고, 사무처리에 적합한 언어가 있는가 하면 시스템프로그래밍에 적합한 언어 등이 있어서 전문 프로그래밍 작업인 경우 그 특성에 따른 언어를 선택해야 하는 것이다.

필자가 생각하는 바로는 전문프로그래머가 아닌 경우, 배우기 쉽고 웬만한 기능들이 골고루 갖춰진 베이직언어가 퍼스널컴퓨터에서는 아주 적합하리라고 본다.

베이직(BASIC : Beginner's All-purpose Symbolic Instruction Code) 언어는 1964년에 미국의 케머니(John Kemeny)와 쿠르츠(Thomas Kurtz)에 의해 개발되어 원래 대형컴퓨터의 단말기에서 사용되도록 되었었다. 케머니와 쿠르츠가 다트무스(Dartmouth)대학에서 컴퓨터를 강의할 때 자연계뿐 아니라 인문계의 학생들도 쉽게 컴퓨터를 사용할 수 있도록 배려하여 베이직을 만들었으므로 초보자가 배우기에는 가장 적당한 컴퓨터언어인 것이 당연하다.

## 16비트 PC와 베이직

원래 대형컴퓨터에서 개발된 베이직이지만 요즈음은 퍼스널컴퓨터에서 그 진가를 발휘하고 있다. 우리나라에서는 처음에 8비트 퍼스널컴퓨터인 애플컴퓨터에서 애플소프트베이직(Applesoft BASIC)이 많이 사용되었었고, CP/M을 사용하는 8비트 컴퓨터에서는 엠베이직(MBASIC)이 많이 사용되어 왔다. 애플컴퓨터에서는 한글을 사용하는 프로그램(콜3327 한글이라 불린다)이 일찍 개발되어 한글을 사용한 애플베이직 프로그램이 널리 사용될 수 있었다.

영문자는 가로 40자, 세로 24행을 한 화면에 보여줄 수 있으나, 콜3327 한글은 가로로 20자, 세로로 12행밖에 화면에 나타낼 수 없었다. 이런 시기에 유저의 답답한 심정을 이해하듯 한글 바이텍스카드가 개발되어 한글도 영문자만큼 한화면에 표시하게 되었다. 그러나 한글 바이덱스카드를 만들면서 애플베이직의 명령과 다른 명령을 추가시킴으로써, 비록 기능은 향상되었으나 기존 프로그램을 그대로 사용하지 못하고 일부 명령어를 수정해야하는 불편을 초래하여 예측과는 달리 널리 쓰이지 못하였다.

바로 이즈음에 16비트 퍼스널컴퓨터(IBM-PC XT)가 국내에 상륙하여 애플컴퓨터에 한글 바이덱스카드를 추가하는 것 보다는 16비트 퍼스널컴퓨터를 구입하는 것이 실용적이 되었고, 동시에 그 유명하던 애플베이직도 IBM 베이직에 선두자리를 물려주게 된다.

더구나 요즈음에는 IBM베이직언어도 BASIC, BASICA 뿐 아니라 성능이 더 우수한 것이 많이 개발되고 있다. TRUE BASIC, BETTER BASIC, QUICK BASIC을 비롯하여 IBM-PC XT의 허큘리스 보드에서 쓸 수 있는 HBASIC같은 것이 계속 발표되고 있으며, 1987년 봄에는 TURBO BASIC*이 미국 볼랜드사에서 개발되어 베이직언어의 수준이 더욱 높아가고 있다.

## 베이직 명령어

시중에 많이 사용되는 IBM베이직 인터프리터에는 BASIC, BASICA, GW BASIC 등이 있으나 특수한 몇개의 명령어를 제외하고는 같은 명령어를 사용한다. 이글을 읽고있는 독자라면 나름대로 베이직 교본(16비트용) 한권쯤은 갖고 있으리라 믿고 여기서는 자주 통용되는 명령어리스트만 **표 1**에 소개한다.

여기서는 명령어해설 수준에 머무르고 있는 일반 매뉴얼 설정을 감안하여 실제 유용한 프로그램을 만드는데 자주

* 터보베이직(TurboBASIC)같은 컴파일러방식에서는 문장번호없이 레이블만을 사용하는 경우도 있다. 인터프리터와 컴파일러방식의 차이점은 다음 연재에서 소개할 예정이다.

〔표 1〕 중요한 IBM 베이직 명령어 및 논리연산자 요약

| | | |
|---|---|---|
| ABS | AND | ASC |
| ATN | BEEP | CALL |
| CHR$ | CINT | CIRCLE |
| CLEAR | CLEAR, 32768 | CLS |
| COLOR | CONT | COS |
| CSRLIN | CVD, CVI, CVS | DATA |
| DATE$ | DEFDBL, DEFINT | DEFSNG, DEF FN |
| DEFSTR, DEFUSR | DELETE | DIM |
| DRAW | ELSE | END |
| ERASE | ERL | ERR |
| ERROR | FILES | FOR…TO…NEXT |
| FRE | GET | GOSUB |
| GOTO | HEX$ | IF…GOSUB |
| IF…THEN | INKEY$ | INPUT, LINE INPUT, INPUT$ |
| INSTR | INT | KILL |
| LEFT$ | LEN | LINE |
| LIST | LLIST | LOAD |
| LPRINT, LPRINT USING | LOCATE | MERGE |
| MID$ | MKD$, MKI$, MKS$ | NAME…AS |
| NEW | NOT | ON COM(1) GOSUB |
| ON ERROR…GOTO | ON…GOSUB | ON…GOTO |
| ON KEY(1) GOSUB | OPEN | OPEN FOR APPEND |
| OPTION BASE | OR | PAINT |
| PEEK | PLAY | POINT, PSET |
| POKE | POS | PRINT, PRINT USING |
| PUT | RND, RANDOMIZE | READ |
| REM 또는 ' | RESTORE | RETURN |
| RIGHT$ | RUN | SAVE |
| SCREEN | SPC | STEP |
| STOP | STORE | STR$ |
| STRING$ | SWAP | SYSTEM |
| TAB | TIME$ | TROFF |
| TRON | USR | VAL |
| WHILE…WEND | WIDTH | XOR |

쓰이고 또 중요한 명령어에 한해 수시로 다루었다.

# 베이직 기초

## (1) 명령어 입력모드

베이직 명령을 입력하는 방법에는 직접모드방식과 간접모드방식, 두가지가 있다. 직접모드방식은 일련의 명령어를 입력하여(명령어가 2개 이상일 때는 ':'으로 분리) 〈ENTER〉키를 누르면 직접 실행되는 방식이다. 그러나 이 경우, 한번 실행된 후에는 다시 시행시킬 수가 없게 된다. 왜냐하면 입력된 명령어들이 기억되어 있지 않기 때문이다. 이에 반해 간접모드방식에서는 입력된 명령어들이 메모리에 기억되어지므로 언제든지 'RUN'을 입력하여 실행시킬 수 있게 된다. 이런 이유로 이 방식을 프로그램모드라고도 하며 보통 프로그래밍작업이라고 하는 것은 이 방식하의 작업을 말한다. 그리고 이 모드에서는 입력되는 명령어를 기억시키기 위해(또한 명령어 실행 순서를 제어하기 위해) 반드시 문장번호(1-65529)를 지정한 후 명령어를 입력하여야 하며 'RUN'(또는 함수키 F2)를 입력하여 실행시켜야 한다.

**그림 1**은 직접모드방식에서의 명령어처리 예를 보여주고 있다. 한번 실행 후에 〈ENTER〉키 또는 'RUN'을 계속 입력해도 더이상 실행되지 않는다.

**그림 2**에서는 간접모드방식(프로그램모드방식)에서의 예를 보인 것이다. 이때는 프로그램이 기억되어 있는 상태이기 때문에 수시로 'RUN'을 이용하여 실행을 시킬 수 있다.

## (2) 기본 명령어

베이직언어에 예약된 명령어는 인터프리터마다 다르지만 대개 약 100개정도이다. 여기서 실무에 주로 사용되는 것은 약 50개에 불과하다. 이같이 몇개 안되는 명령어로 프로그램을 작성할 수 있는 점이 베이직언어의 장점이다. 쉽게 배울 수 있고, 바로 응용하여 프로그램을 작성할 수 있는 대화식 컴퓨터 언어이기 때문에 수많은 사람들에게서 이용되고 있는 것이다. 우선 15개 정도만 여기서 살펴보기로 하자.

- **프로그램 제어용 명령어** : LIST, NEW, RUN, END, SYSTEM
- **화일 제어용 명령어** : LOAD, SAVE, FILES
- **입·출력 명령어** : PRINT, CLS, LOCATE, INPUT
- **조건 및 분기 명령어** : (IF…) GOTO, (IF…) GOSUB
- **반복 명령어** : FOR…NEXT

우선 **그림 3**에 나와 있는 프로그램을 입력해 보자. 한자 한자 틀림없이 입력해야 한다. 보통 프로그램논리에는 이상이 없지만 입력을 잘못해서 문법(syntax) 에러를 유발시키는 경우가 많다. 잘못 입력된 부분이 발견되면 화살표키로 움직여 수정한 후 〈ENTER〉키를 누르면 된다.

입력이 끝났으면 다음과 같이 화일제어용 명령어 'SAVE'를 이용하여 디스켓(또는 하드디스크)에 프로그램을 저장해 둔다.

```
Ok
SAVE "TEST-1"
Ok
```

〔그림 1〕 직접모드에서의 프로그램 실행

```
Ok
PRINT " 이 기성 "
이 기성
Ok
PRINT 100+300
 400
Ok
?450*100
 45000
Ok
CLS:PRINT:PRINT "***** 직접모드에서의 입력테스트 *****"


***** 직접모드에서의 입력테스트 *****
Ok
```

〔그림 2〕간접모드에서의 프로그램 실행

```
Ok
10 CLS
20 PRINT: PRINT
30 PRINT "***** 간접모드에서의 입력테스트 *****"
40 BEEP
50 PRINT "1988년 12월 ..."

RUN


***** 간접모드에서의 입력테스트 *****
1988년 12월 ...
OK
```

〔그림 3〕 TEST-1. BAS 리스트

```
10 REM TEST-1.BAS 프로그램 ------------------------------
20 CLS: Y = 10: X = 25
30 LOCATE Y,X   : PRINT "***************************"
40 LOCATE Y+1,X : PRINT "*                         *"
50 LOCATE Y+2,X : PRINT "*         안녕하세요        *"
60 LOCATE Y+3,X : PRINT "*                         *"
70 LOCATE Y+4,X : PRINT "***************************"
80 BEEP
```

자, 이제 방금 작성한 프로그램이 보조기억장치에 기억되었다. 아직 주기억장치에도 여전히 기억된 상태로 있기 때문에 RUN명령어로 실행시킬 수 있다. 화면상에 'RUN'을 입력하여 실행시키면 **그림 4**와 같이 화면 중앙부분에 박스(BOX)로 둘러싸인 메시지가 나온다.

만약 에러가 발생하면 잘못된 문장번호 또는 그 문장내용이 화면에 나타나므로 이를 올바르게 수정한 후 같은 화일이름으로 다시 세이브한다. 그러면 기존의 화일이름으로 수정된 프로그램이 기억된다.

그림 3의 프로그램을 보면 처음에 CLS 명령으로 화면이 깨끗이 지워진 후 LOCATE 명령으로 지정된 화면위치에, PRINT 명령으로 지정해준 메시지를 그대로 출력시켜 주고 있음을 볼 수 있다. 이 짧은 예제프로그램을 가지고 다음 명령어들을 각자가 테스트해 보기 바란다.

- **FILES** : 보조기억장치에 있는 화일목록들을 보여준다(방금 입력한 TEST-1.BAS가 보일 것이다).
- **LIST** : 현재 주기억메모리에 들어있는 프로그램의 내용을 화면에 보여준다. 만약 프린터로 이 리스트를 출력시키려면 LLIST명령을 사용한다.
- **CLS** : 주기억메모리와는 관계없이 화면만을 깨끗이 지우고 커서를 맨 좌측 상단으로 옮긴다.
- **NEW** : 화면과는 관계없이 주기억메모리에 기억되어 있는 내용을 모두 지운다. 그러나 이미 보조기억장치(플로피나 하드디스크)에 기억된 상태이므로 다시 이 프로그램을 실행시키려면 LOAD 명령을 사용하면 된다.
- **LOAD** : 보조기억장치에 있는 프로그램을 주기억장치로 옮겨준다(컴퓨터는 주기억장치에 있는 내용만을 처리한다).
- **SYSTEM** : 베이직을 끝내고 다시 도스상태로 돌아간다.

다음은 조건 및 분기명령을 이용하여 그림 4의 메시지를 화면상에서 이동시켜 보자. 프로그램을 잘 살펴보면 컴퓨터가 얼마나 단순하게 처리하고 있는지를 금방 알 수 있다. 여기서 단순성, 고속성, 정확성이란 컴퓨터특성 한가지를

〔그림 4〕 TEST-1.BAS 결과

```
***************************
*                         *
*        안녕하세요        *
*                         *
***************************
Ok
```

소개하고 싶다. 즉 컴퓨터는 아주 단순한 처리를 고속으로 정확하게 실행시켜 주기 때문에 복잡하고 섬세한 업무도 단숨에 처리하고 있는 것처럼 보이는 것이다.

**그림 5**는 그림 3의 프로그램을 메시지를 만드는 하나의 서브루틴으로 보고 화면위치를 원하는 곳에 지정하여 그때마다 이 루틴을 불러내는게 이 프로그램의 내용이다.

그림 5의 MOVE1.BAS을 입력하고 독자들 나름대로 수정하여 그 출력효과를 살펴보기 바란다.

- **LOCATE 〈행〉,〈열〉**: 화면의 좌표를 지정한다.
- **GOSUB 〈문장번호〉**: 서브루틴을 수행한 후 돌아온다. 〈문장번호〉는 서브루틴이 시작되는 문장번호를 나타내고 서브루틴은 반드시 'RETURN'으로 끝내야 한다.

이쯤되면 독자들 중 프로그램 기질(?)이 다분히 있는 사람은 출력메시지를 어떤 규칙하에서 보기좋게 이동시켜보고 싶은 욕심이 금방 생길 것이다. 이를 위해 반복명령어 'FOR…NEXT'가 준비되어 있다. 이 명령어는 베이직프로그래밍에서 약방의 감초같이 자주 쓰이는 매우 유용한 것이므로 잘 익혀두기 바란다.

여기서는 다음과 같이 화면이 가로, 세로 방향으로 이동시키는 예를 보였으나 대각선방향 및 이동되는 자취를 남기기(행번호 90의 CLS명령을 삭제하면 됨) 등 메인 루틴을 변형함에 따라 여러가지 출력효과를 낼 수 있다.

참고로, 명령어도 익히고 프로그램의 묘미도 느낄겸 입·출력 명령어인 'INPUT'와 조건·분기 명령어인 'IF' 명령의 쓰임새도 첨가했다.

**그림 6**에 MOVE2.BAS 리스트를 나타냈다.

- **INPUT "프롬프트";〈변수〉**: 프로그램실행 도중 키보드로부터 입력을 받아들인다. 여기서 "프롬프트"는 원하는 입력의 메시지에 사용되는 스트링상수이고 〈변수〉는 입력을 받아들이는 변수명으로 여럿이 올 수 있다.
- **IF 〈식〉 THEN 〈처리1〉 ELSE 〈처리 2〉**: 식의 결과에 따라 프로그램 흐름을 원하는 곳으로 분기한다. 〈처리1〉, 〈처리2〉는 처리할 문장번호 또는 직접 처리명령을 나타낸다. ELSE 이하는 생략가능하며 생략시 다음 문장을 지정한다.
- **BEEP**: 스피커로 '삑'소리를 낸다.

## (3) 알아두면 편리한 기능

IBM PC계열의 키보드를 보면 왼쪽에 F1에서 F10까지 적혀진 10개의 키가 있다. 이 키들은 도스에서 템플리트 기능을 갖는가하면 그외의 각종 소프트웨어에서 특수한 기능들을 갖고 실행되므로 함수키(Function Key)라 불리운다. GWBASIC같은 경우도 이 10개의 함수키 모두가 유용한 기능으로 쓰이고 있기 때문에 프로그래밍작업시 상당한 효율성을 가져다 줄 수 있다. **그림 7**은 GWBASIC을 실행시킨 초기화면을 나타낸 것이다. 화면 밑부분을 보면 이들 함수키의 기능을 설명한 메시지가 보인다. 이들 함수키 기능들은 소프트적으로 할당된 초기값이므로 프로그램내에서 재정의도 가능한데 이는 차후에 설명하고 일단 여기서는 초기값만을 익히기로 하자.

초보자인 경우 먼저, 가장 많이 쓰이는 F1에서 F4까지의 기능만을 꼭 익혀두기 바란다. F3키를 눌러보라. 화면에 "LOAD"라는 메시지가 나온다. 여기에 MOVE2를 입력하고 〈ENTER〉키를 누른다. OK 메시지가 나오면 이번엔 F1키를 눌러 〈ENTER〉키를 누른다. 어느새 MOVE2.BAS 프로그램

〔그림 5〕 MOVE 1.BAS 리스트

```
10 REM MOVE1.BAS 프로그램 ------------------------------
20 CLS
30 BEEP:LOCATE 1,1:PRINT "메시지를 표시할 좌표를 입력하세요 "
40 LOCATE 2,1:INPUT "X 좌표를 입력하세요 ( 0 <= X => 55) ";X
50 LOCATE 3,1:INPUT "Y 좌표를 입력하세요 ( 4 <= Y => 20) ";Y
70 IF X<0 OR X>55 THEN BEEP: GOTO 20
80 IF Y<0 OR Y>20 THEN BEEP: GOTO 20
90 CLS:LOCATE Y,X: GOSUB 100
95 GOTO 30
100 '-------- 메시지를 그리는 부분 --------------------
110 LOCATE Y,X   :PRINT "***************************"
120 LOCATE Y+1,X :PRINT "*                         *"
130 LOCATE Y+2,X :PRINT "*        안녕하세요        *"
140 LOCATE Y+3,X :PRINT "*                         *"
150 LOCATE Y+4,X :PRINT "***************************"
160 RETURN
```

〔그림 6〕 MOVE 2.BAS 리스트

```
10 'MOVE2.BAS 프로그램
20 CLS: Y = 10: X = 25
30 GOSUB 600 '초기 화면을 그린다
40 GOSUB 100 '메시지를 좌로 이동
50 GOSUB 200 '메시지를 위로 이동
60 GOSUB 300 '메시지를 우로 이동
70 GOSUB 400 '메시지를 아래로 이동
80 GOSUB 500 '메시지를 중앙으로 이동
90 GOSUB 700 '최종화면을 그린다
95 END
100 '-----------------------------------------
110 FOR I = 1 TO 12
120 GOSUB 600
130 X = X - 2
140 CLS
150 NEXT I
160 BEEP: RETURN
200 '-----------------------------------------
210 FOR J = 1 TO 4
220 GOSUB 600
230 Y = Y -2
240 CLS
250 NEXT J
260 BEEP: RETURN
300 '-----------------------------------------
310 FOR K = 1 TO 27
320 GOSUB 600
330 X = X + 2
340 CLS
350 NEXT K
360 BEEP: RETURN
400 '-----------------------------------------
410 FOR L = 1 TO 5
420 GOSUB 600
430 Y = Y + 2
440 CLS
450 NEXT L
460 BEEP: RETURN
500 '-----------------------------------------
510 FOR M = 1 TO 15
520 GOSUB 600
530 X = X - 2
540 CLS
550 NEXT M
560 RETURN
600 '-----------------------------------------
610 LOCATE Y,X   : PRINT "**********************"
620 LOCATE Y+1,X : PRINT "*                    *"
630 LOCATE Y+2,X : PRINT "*      안녕하세요      *"
640 LOCATE Y+3,X : PRINT "*                    *"
650 LOCATE Y+4,X : PRINT "**********************"
660 RETURN
700 '-----------------------------------------
710 CLS: Y = 10:X = 25
720 GOSUB 600
730 FOR F = 300 TO 500 STEP 100
740 SOUND F,2
750 SOUND 32767,2
760 NEXT F
770 LOCATE 16,23: PRINT "화면을 다 돌았읍니다."
780 LOCATE 20,23: COLOR 0,7: INPUT "계속 실행할까요(Y/N)";A$
   :COLOR 7,0
790 IF A$ = "" THEN GOTO 780
800 IF A$ = "N" OR A$ = "n" THEN 820
810 GOTO 10
820 RETURN
```

〔그림 7〕 GWBASIC의 초기화면

```
GW-BASIC 2.01
(C) Copyright Microsoft 1983,1984
Copyright 1984 TeleVideo Systems, Inc.
62819 Bytes free
Ok









1LIST  2RUN←  3LOAD"  4SAVE"  5CONT←  6,"LPT"  7TRON←  8TROFF←  9KEY  0SCREEN
```

이 로드되어 리스트가 디스플레이된다. 이제 단지 F2키를 누름으로써 실행을 시킬 수 있다.

다음은 AUTO 명령어에 대해서 알아보자. 이 명령은 지정된 간격으로 문장번호를 자동적으로 지정해 준다. 프로그램을 입력할 때 매번 문장번호를 입력한다는 것은 꽤나 귀찮은 작업이었다. 이제부터는 AUTO 명령으로 프로그램을 입력하기 바란다.

● **KEY ON/OFF** : 화면의 25번째 행에 함수키 메시지를 지우거나 표시할 수 있다.

●AUTO 〈시작문번호〉, 〈간격〉:〈ENTER〉 키를 누를 때마다 자동적으로 문장번호를 만들어 준다. 〈시작문번호〉와 〈간격〉을 생략하면 디폴트로 각각 10이 지정된다.

# 프로그래밍 테크닉

## (1) 순서도

베이직언어의 많은 장점에도 불구하고 한가지 단점이 있다면 바로 비논리적인 프로그램구성으로 흐르기 쉽다는 점이다.

그럼 구조적프로그래밍(Structured Programming)이란 어떤 것인가? 이에 대해서는 많은 이견이 있을 수 있지만 요약해서 보면 전반적인 프로그램논리가 각 기능별로 모듈화된 부분 사이를 순차적으로 진행해 가면서 정확한 판단조건에 의한 분기와 반복을 유지하며 막힘없이 흘러가는 구조를 말한다. COBOL이나 PASCAL같은 구조적언어에서는 그리 심각한 문제가 아니나 베이직에서는 그렇지 못하기 때문에 처음부터 프로그램을 구상할 때 순서도를 미리 그려보는 습관을 꼭 지니기 바란다.

그리고 프로그램 작성시 되도록이면 GOTO(무조건 분기명령) 보다는 IF…THEN…ELSE 명령이나 WHILE…WEND 및 GOSUB 명령어를 이용하고 FOR…NEXT같은 반복명령을 효율적으로 이용하는 방법을 택하는 것이 좋다. 표 2에 간단한 논리구조 순서도를 보였다.

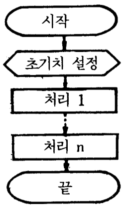
●순서도(Flow Chart) : 어떠한 일의 순서나 판단에 따른 절차 등을 기호로 알기 쉽게 도시한 것이다. 주로 프로그램 작성시 필요하며 전반적인 프로그램 흐름을 약속된 기호로 표시한다.

## (2) 차림표에 의한 모듈처리

일반적으로 사용자가 쉽고 편하게 이용할 수 있는 프로그램이 좋은 프로그램에 속한다. 즉 화면에 각종 처리사항 및 관련정보, 도움말 등을 보기좋게

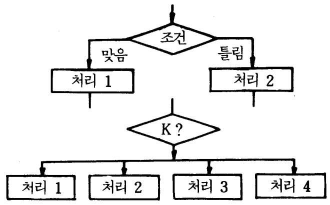
〔표 2〕 간단한 논리구조

**1. 순차처리형**




```
10 CLS
20 INPUT "데이터-1"; X
30 INPUT "데이터-2"; Y
40 TOT = X + Y
50 AVE = TOT/2
60 PRINT "합  =   ";TOT
70 PRINT "평균 =   ";AVE
80 END
```

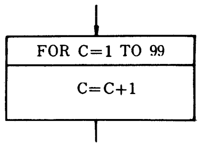
**2. 조건분기**




```
.
.
50 IF A > B THEN PRINT "A가 큽니다."
60 PRINT "B가 크거나 같습니다."


.
.
50 ON K GOSUB 100, 200, 300, 400
```

**3. 반복형**




```
10 FOR C = 0 TO 99
20     C = C + 1
30 NEXT C
40 PRINT C

10 C = 0
20 WHILE C < 100
30   C = C + 1
40 WEND
50 PRINT C
```

〔그림 8〕 SAMPLE.BAS 리스트

```
10 'Sample.bas 프로그램
20 '-------------- 메인 루틴 ---------------------------------------------
25 GOSUB 200 '상수를 초기화시킨다.
30 GOSUB 300 '화면 차림표를 만든다.
40 GOSUB 430 '선택을 체크한다.
50 IF OK$ = "FALSE" THEN END
60   ON Q% GOSUB 1000, 2000, 3000, 4000, 5000
70 GOTO 30
200 ' ------------ 상수 초기화 ------------------------------------------
210 DIM DAT(500)
220 SPAS1$ = "                    ": OK$ = "TRUE"
230 FOR I = 1 TO 500
240 DAT(I) = 0
250 NEXT I
260 RETURN
300 ' ----------- 초기화면 구성 -----------------------------------------
310 CLS
320 LOCATE 3,18:COLOR 0,7:PRINT "*** 예제 프로그램 ***":COLOR 7,0
330 PRINT: PRINT TAB(55) "1988년 12월"
```

```
340 PRINT "============================================================"
350 LOCATE 10,10: PRINT "1. 데이터처리를 위한 새화일 만들기  "
360 LOCATE 11,10: PRINT "2. 자료입력, 수정하기               "
370 LOCATE 12,10: PRINT "3. 자료찾아보기                     "
380 LOCATE 13,10: PRINT "4. 자료삭제하기                     "
390 LOCATE 14,10: PRINT "5. 도움말                           "
400 LOCATE 15,10: PRINT "Q. 작업종료                         "
410 LOCATE 20,15: PRINT "선택 하십시요 ";:BEEP
420 RETURN
430 '----------- 선택 루틴 ------------------------------------------
440 Q$ = INPUT$(1)
450 IF Q$ = "Q" OR Q$ = "q" THEN OK$ = "FALSE"
460 Q% = VAL(Q$)
470 RETURN
1000 '--------- 새로운 화일만들기 ------------------------------------
1010 CLS
1020 PRINT: PRINT
1030 PRINT TAB(5) "도움말 : 이곳은 성적처리를 위한 새화일을 만드는
     곳입니다."
1040 PRINT TAB(5) "      코드번호, 학년, 반번호를 입력하세요.  "
1050 PRINT "============================================================"
1060 PRINT: PRINT: PRINT
1070 LOCATE 10,5:COLOR 0,7:INPUT "코드번호를 입력하세요
     :";CODE%: COLOR 7,0
1080 LOCATE 11,5:COLOR 0,7:INPUT "학년번호를 입력하세요
     :";GRAD%: COLOR 7,0
1090 LOCATE 12,5:COLOR 0,7:INPUT "학급번호를 입력하세요
     :";CLAS%: COLOR 7,0
1100 IF STR$(CODE%)<> "" AND STR$(GRAD%)<> ""
     AND STR$(CLAS%)<> "" THEN 1160
1110 BEEP:LOCATE 15,5:PRINT "입력이 잘못됐읍니다."
1120 FOR I = 10 TO 12
1130 LOCATE I,27:PRINT SPAS1$
1140 NEXT I
1150 GOTO 1070
1160 LOCATE 23,10: PRINT "아무키나 누르면 계속됩니다."
1170 ANS$ = INKEY$: IF ANS$ = "" THEN 1170
1180 RETURN
2000 '--------- 자료입력하기 ----------------------------------------
2010 BEEP: CLS: PRINT: PRINT
2020 PRINT TAB(5) "도움말 : 여기서는 자료를 입력하거나 수정할 수
     있읍니다."
2030 PRINT TAB(5) "      입력할 자료수를 입력하세요              "
2040 PRINT "============================================================"
2050 PRINT
2060 INPUT "             입력할 자료수는 ";NUM$
2070 IF VAL(NUM$) = 0 THEN 2150
2075 NUM% = VAL(NUM$)
2080 FOR COUNT = 1 TO NUM%
2085 LOCATE 8,17:PRINT SPAS1$:LOCATE 8,10
2090 COLOR 0,7:PRINT "      ";COUNT;" = ";
     :INPUT DAT(COUNT):COLOR 7,0
2140 NEXT COUNT
2150 RETURN
3000 '--------- 자료찾아보기 ----------------------------------------
3010 CLS
3020 PRINT: PRINT
3030 PRINT TAB(5) "도움말 : 자료를 찾아보기 위한 곳입니다."
3040 PRINT TAB(5) "      원하는 기능을 선택하세요."
3050 PRINT "============================================================"
3060 PRINT: PRINT: PRINT
3070 PRINT TAB(5) "1. 데이터자료보기        ":PRINT
3080 PRINT TAB(5) "2. 화일목록보기          ":PRINT
3090 PRINT TAB(5) "=================="
3100 INPUT "          원하는 기능번호는 ";FUN$
3110 IF FUN$ = "" THEN 3140
3120 ON VAL(FUN$) GOSUB 3200, 3400
3130 GOTO 3000
3140 RETURN
3200 CLS
3210 FOR COUNT = 1 TO NUM%
3220  PRINT DAT(COUNT)
3230 NEXT COUNT
3240 LOCATE 23,5: PRINT "아무키나 누르면 계속됩니다....."
3250 ANS$ = INKEY$: IF ANS$ = "" THEN 3250
3300 RETURN
3400 CLS
3410 FILES
3420 LOCATE 23,5: PRINT "아무키나 누르면 계속됩니다....."
3430 ANS$ = INKEY$: IF ANS$ = "" THEN 3430
3450 RETURN
4000 '----------- 자료삭제하기 --------------------------------------
4010 CLS
4020 PRINT: PRINT
4030 PRINT TAB(5) "도움말 : 자료를 삭제하는 루틴입니다."
4040 PRINT TAB(5) "      여기서는 화일목록만을 지우는 것으로
     대신합니다."
4050 PRINT "============================================================"
4060 PRINT: PRINT
4070 LOCATE 8,8:COLOR 0,7:INPUT "삭제할 화일을 입력하세요 "
     ";FILE$:COLOR 7,0
4080 IF FILE$ = "" THEN 4130
4090 FIL$ = FILE$+".BAS"
4100 KILL FIL$
4110 LOCATE 10,8: PRINT "이제 화일 "FIL$+"이 지워졌읍니다."
4120 FOR TIME = 1 TO 3000:NEXT TIME
4130 RETURN
5000 '----------- 도움말 ------------------------------------------
5010 CLS
5020 PRINT: PRINT: PRINT
5030 PRINT TAB(5) "이 프로그램은 실제 자료처리보다는 전반적 프로그램 "
5040 PRINT TAB(5) "구조를 설명하기 위한 것이므로 여기서 데이터화일들 "
5050 PRINT TAB(5) "은 가상적으로 처리되고 있읍니다. "
5060 PRINT TAB(5) "여기서 소개된 각 루틴은 독자적인 화일단위로도 이용"
```

```
5070 PRINT TAB(5) "될 수 있읍니다. 다음 연재에서 그 원리를 다루겠읍니다."
5080 LOCATE 23,5: PRINT "아무키나 누르면 계속됩니다....."
5090 ANS$ = INKEY$:IF ANS$ = "" THEN 5090
5100 RETURN
```

〔그림 9〕 SAMPLE.BAS 초기화면

```
        *** 예제 프로그램 ***

                                   1988년 12월
=====================================================================

    1: 데이터처리를 위한 새화일 만들기
    2. 자료입력, 수정하기
    3. 자료찾아보기
    4. 자료삭제하기
    5. 도움말
    Q. 작업종료

        선택 하십시요
```

표시하여 사용자가 직접 보면서 단계별로 처리를 할 수 있게 프로그램을 만들면 좋다. **그림 8**에 간단한 예를 보였다. 이 프로그램은 살을 모두 떼어낸 매우 간략화시킨 예이기 때문에 독자들이 쉽게 수정을 가하며 학습할 수 있으리라 본다. 그리고 이 프로그램에서 각 모듈들이 방대해질 때에는 각각 독자적인 화일로써 처리시킬 수도 있다. 디스크화일처리 및 그 이용법에 대해서는 다음 연재에서 자세히 다룰 것이다.

- **COLOR 7, 0**: 정상상태의 화면을 지정한다.
- **COLOR 0, 7**: 화면이 역상으로 된다.
- **TAB**: PRINT 명령과 함께 사용되며 화면의 맨 왼쪽부터의 칸수를 나타낸다.
- **INKEY$**: 키보드로부터 문자를 읽어들인다.
- **INPUT$(n)**: 키보드 또는 화일로부터 문자스트링을 구한다. 여기서 n은 읽혀질 문자의 수를 나타낸다.
- **VAL**: 문자열을 숫자값으로 바꾸어주는 명령이다. VAL뒤의 문자열이 숫자가 아닌 문자라면 그 숫자값은 0가 된다.
- **ON…GOSUB**: ON 다음의 숫자값에 해당하는 순서의 GOSUB을 수행한다.

**그림 9**에 SAMPLE.BAS 프로그램 초기화면을 나타냈다.

ON…GOSUB 명령으로 분기되어 있는 각 루틴들을 독자들이 더욱 보강하여 유용한 프로그램이 탄생되길 기대해 본다. PC

# dBASE Ⅲ+의 개요

박성현/공영종합시스템 개발부

DBASE Ⅲ라는 프로그램은 PC에 조그마한 관심을 가지고 있는 사람이라면 모르는 사람이 없을 정도로 많이 알려져 있는 프로그램이다. 조그마한 시간적 여유가 없어 디베이스를 사용하기 어렵다고 말하는 사람이나 접해보지 않은 사람일지라도 디베이스를 연구해 보면 아주 쉽게 사용할 수 있다는 것을 알 수 있으며 그 기능에 감탄을 할 것이다.

디베이스야말로 PC사용자를 모두 프로그래머로 만들 수 있는 도구라고 자신있게 말할 수 있을 정도로 그 사용이 간단하고 강력한 기능을 가졌다고 할 수 있겠다.

dBASE의 달인 아담 그린*(Adam b. Green)은 컴퓨터 강의시 마다 "네 컴퓨터에게 누가 주인인지 가르쳐 주라"고 한다. 이에 디베이스 강좌를 맡으면서 이글을 읽는 독자 모두가 디베이스를 통하여 독자의 PC에게 독자가 주인이라는 걸 가르쳐 줄 수 있도록 하고자 하며 또한 디베이스야 말로 내 컴퓨터의 모든 것이라는 생각이 들 정도의 매력있는 것이 되도록 만들어 보고자 한다.

아울러 이 글을 읽는 모든 독자가 자신이 필요한 프로그램은 직접 작성할 수 있는 능력을 갖출 수 있도록 하고자 한다. 아직 프로그램을 할 줄 모르는 사람이라도 이 디베이스 강좌를 통해 한번 디베이스에 푹 빠져 보길 바란다. 아담 그린과 같이 그리고 필자와 같이 인생이 바뀔줄도 모르는 사태(?)가 일어날지 모른다.

## 디베이스의 역사

디베이스의 첫번째 제품은 디베이스 Ⅰ이 아니라 처음부터 Ⅱ를 사용함으로써 새로운 버전의 프로그램인 것처럼 하는 마케팅 전략으로 Ⅱ를 사용한 것임을 먼저 알려준다. 디베이스의 탄생은 다음과 같다.

1970년대 초반에 미국 캘리포니아(California)주 패사데나(Pasadena)에 있는 제트 추진 연구소(JPL:Jet Propulsion Laboratory)에서 무인탐사 인공위성으로 부터 전송받은 방대한 양의 정보를 분석하기 위하여 Jeb Long이라는 사람이 프로그래밍한 JPLDIS(JPL Data Management and Information Retrival System:제트 추진 연구소용 데이터관리 및 정보회수 시스템)에 감명을 받은 웨인 레트립(C.Wayne Ratliff)이 이와 비슷한 프로그램을 만들어 보려고 자신의 마이크로 컴퓨터에서 그 시도를 하게 되었다.

1979년 그는 CP/M을 기초로 한 이 프로그램을 발칸(Vulcan)이라는 이름으로 판매를 시도하였으나 실패를 거듭한다. 이때 이 프로그램의 기능을 본 죠지 테이트(Jeorge Tate, 소프트웨어 판매회사 경영)가 많은 액수의 저작 사용료를 지불하고 이 프로그램을 팔기 시작했다. 이때부터 발칸의 이름에는 dBASE Ⅱ라는 이름이 붙어 다녔고 1984년에는 1개월당 2,000개씩의 카피본이 팔렸고 아쉬톤 테이트라는 이 프로그램을 팔기 위한 특정한 회사를 설립하기에 이르렀으며 마이크로 소프트(MicroSoft), 마이크로프로 인터내셔날(MicroPro International) 등의 회사와 어깨를 견줄 정도가 되었다.

디베이스 Ⅱ가 가진 가장 놀라운 사실은 자체의 프로그램 랭귀지를 가지고 있다는 사실이다. 디베이스는 모든 명령을 직접 키보드를 통하여 내릴 수 있는데 한번의 명령을 통하여 원하는 데이터를 마음대로 이용할 수가 있다. 이러한 명령을 하나의 화일에 묶어 IF와 LOOP로 연속적으로 자동처리를 한다면 얼마나 간편하고 쉬운 랭귀지인가 생각해보라. 디베이스는 타 랭귀지로 수백, 수천 라인에 해당하는 것을 단 한줄의 명령으로 표현할 수 있다. 이렇게 복잡한 프로그램이 단 한줄로 표현된다면 얼마나 쉽게 프로그램할 수 있겠는가 각자가 상상해보라.

프로그램 지식이 없는 사람이 가장 쉬운 방법으로 프로그래밍을 할 수 있다는 매력이 디베이스를 성공하게 한 것이라 보며 실제로 누구라도 프로그램을 작성할 수 있도록 되어 있다(지금 이글을 읽는 독자가 이글의 연재를 끝까지 읽는다면 MS-DOS만 알고 있는 한, 만약 MS-DOS를 모른다면 함께 공부해 가며 읽는다면 분명 자신이 필요한 프로그램은 직접 자신이 작성할

***아담 그린(Adam b. Green)**
dBASE 초기 사용자중의 한사람으로서 그와 dBASE Ⅱ의 만남은 그의 일생을 온통 뒤바꿔 놓았다. 그의 디베이스와 로터스 강좌는 미국에서 큰 인기를 얻고 있다.

수 있음은 물론이고 친구의 생일 선물로 친구의 이름이 들어가 있는 주소록 프로그램을 선물할 수도 있을 것이다. 자신의 이름이 들어가 있는 프로그램을 받는다면 얼마나 기쁘겠는가 생각해 보라).

이러한 막강한 기능때문에 아쉬톤 테이트는 많은 돈을 벌었지만 이에 대항하는 프로그램이 하나씩 하나씩 나타나기 시작했다. 이에 IBM-PC의 장점을 충분히 이용하고 디베이스 Ⅱ의 문제점을 해결한 새로운 프로그램이 개발되었다. 이것이 디베이스 Ⅲ이며 Ⅱ와 최대한 호환을 가지도록 한 것이다. Ⅱ에서 놀란 사람은 Ⅲ에서는 기절할 정도의 기능을 갖추었다고 할 정도면 이해가 갈 것이다(실로 퍼스널 컴퓨터가 그 정도까지 할 수 있을까 하며 의아하게 생각한 사람이라면 충분히 그러리라 믿는다).

Ⅱ에서는 1레코드당 최대 1,000자 32개의 필드만을 사용할 수 있었으며 레코드의 최대 숫자로는 65,535개를 넘을 수 없었고 소팅은 한개의 필드에 한해 할 수 있었다. 그러나 Ⅲ에서는 1레코드당 4,000자 128개의 필드를 포함할 수 있으며 10억개의 레코드가 한 화일에 들어가며 10개의 데이터 화일을 동시에 열고 7개의 인덱스 화일(소팅된 순서를 유지하도록 해주는 화일이며 데이터가 바뀌면 자동으로 그 순서도 바뀌게 된다)을 동시에 사용할 수 있다. 특히 날짜를 계산할 수 있도록 한 배려로 날짜를 계산하는데 간단한 방법을 쓸 수 있다. 그외로 속도면에서 그리고 기타 사용하기 편리한 기능이 무척 많이 늘어났다.

아담그린은 Ⅲ의 발표와 함께 Ⅱ와 Ⅲ를 비교하여 사용자에게 가르쳐 주는 디베이스를 위한 다리(dBASE Bridge)라는 제목의 글을 쓰게 된다. 그가 이 글을 쓰게 되는 이유는 새로운 사용자가 신버전의 디베이스를 누구보다도 빨리 익히고, 이에 숙달되게 하기 위해서 였으며 디베이스 Ⅳ의 개발 완료와 함께(발표는 아직 안되었음) 선구자로서의 그의 위치는 더욱 확고히 굳었으리라 생각된다.

이후 부터는 디베이스를 더욱 효과적으로 사용할 수 있는 프로그램이 쏟아져 나와 디베이스 유틸리티라는 이름으로 팔리기 시작하면서 디베이스는 한층 순탄대로를 달리게 된다. 그중에서 디베이스 프로그램을 더욱 빨리 실행 시킬 수 있는 디베이스 컴파일러까지 탄생하여 디베이스 자체의 랭귀지가 하나의 프로그램 랭귀지로서 정착을 하게 된다.

PC에 대한 새로운 요구에 맞추어 1985년에는 디베이스 Ⅲ+(PLUS)가 발표됐으며 이는 Ⅲ와 큰 변화는 없으나 Ⅲ를 더욱 간편하게 사용할 수 있도록 하는데 역점을 두었다. 특히 디베이스 Ⅲ+는 LAN(LOCAL AREA NETWORK)을 지원할 수 있도록 하였고 dBCODE와 dBLINKER의 제공으로 프로그램 개발자의 손해를 막았으며, 명령어를 모르는 사람을 위한 ASSIST, HELP를 완벽하게 지원하고 나아가 프로그램 자동생성기인 Application Generator까지 제공되었다.

이렇게 까지 발전하여 사용자의 요구를 충분히 제공하게 되었으나 디베이스 Ⅲ+의 출현은 디베이스 Ⅲ+를 그대로 사용할 수 있으면서도 속도면에서 몇배 빠르고 명령어가 오히려 추가된 제품들이 쏟아져 나오게 되는 계기가 되었다. CLIPPER, FoxBase +, Quick Silver, dBFAST 등의 제품은 디베이스를 지원하는 유틸리티로 출발하여 이제는 디베이스가 필요없이 자체의 제품만으로 충분히 사용할 수 있게 되었으며 디베이스에게 도전장까지 내고 있는 제품이라 할 수 있다. 필자 역시 디베이스는 제쳐놓고 FoxBase +만을 쓰고 있는 실정이다. 바이트지 광고에 의하면 아담 그린 역시 FoxBase +를 찬양하고 있다.

이에 위협을 받은 아쉬톤 테이트는 디베이스 Ⅳ의 개발을 끝내 놓고 그 테스트를 위한 베타 버전을 이미 몇몇 유저에게 나눠 주었다. 디베이스 Ⅳ는 데이터베이스 화일이 만들어 질때 인덱스 화일도 동시에 만들 수 있으며 새로운 인덱스 화일(.MDX)는 한개의(.DBF) 화일에 50개까지 가능 하도록 되어 있다. 물론 수정할 때마다 자동 업데이트 된다.

프로그램 환경에서는 내장 컴파일러에 의해 자동적으로 프로그램, 프로시듀어 및 포맷화일들을 번역하고 분석한다. 프로시듀어들은 어떤 프로그램 화일에도 저장이 가능하고 일단 컴파일되면 어떤 프로그램으로도 호출이 가능하다. 가장 크게 변모한 것은 SQL(Structured Query Language)의 완벽한 수행과 관계형 데이터베이스의 능력 추가라 하겠다. 새로 추가된 명령과 기능은 하이레벨 랭귀지(High Level Language)로서의 자리를 완전히 굳힐 수 있으리라 생각된다.

그러나 아직 판정은 나지 않았다. 과연 디베이스 Ⅳ가 기존의 다른 프로그램을 완전히 K.O시킬 수 있을 것인가? 아니면 영원히 물러서게 될 것인가? (비지캘크가 로터스 1-2-3에게, 로터스 1-2-3은 쿼트로에게 자리를 양보해 주듯이)

필자는 누가 이기든 상관이 없다. 오히려 경쟁을 통해 훌륭한 제품이 나오길 바라는 마음 뿐이며 그런 제품이 우리나라에서 만들어진 제품이길 간절히 원하는 마음이다. 아무리 훌륭한 프로그램이라도 우리나라에서 사용하기에는 한글 사용에 문제점을 안고 있어 100%의 기능을 충분히 활용하기 어려운 실정이다.

## 디베이스의 소개

디베이스는 다음과 같은 디스켓과 매뉴얼로 구성되어 있다.

1. System Disk #1
2. System Disk #1 Backup
3. System Disk #2

4. Sample Program and Utilities Disk
5. On-Disk Tutorial
6. dBASE ADMINISTRATION #1
7. dBASE ADMINISTRATION #2
8. Applications Generator

디베이스는 시스템 디스크 #1과 #2를 사용하게 되며, Sample Program and Utilities Disk는 예제 프로그램과 dBCODE와 dBLINKER 등의 유틸리티가 포함되어 있다. On-Disk Tutorial은 디베이스 교육용 프로그램이며, dBASE ADMINISTRATION #1, #2는 LAN 환경에서 디베이스를 사용하기 위한 프로그램으로 LAN용 소프트웨어가 준비되어 있지 않으면 사용할 수 없다. Application Generator는 프로그램 자동생성기로 간단한 입출력 프로그램을 자동으로 만들어 준다.

아직 데이터베이스라는 것이 무엇인지 모르는 사람이 있다면 위의 디스켓중에 On-Disk Tutorial을 A:드라이브에 넣고

```
A>INTRO ↵
```

를 입력하면 그때부터 데이터베이스의 의미과 dBASE 사용방법을 소개해준다. 물론 영어로 설명되어 이해하기 곤란한 점도 있겠지만 많은 그림이 포함되어 있고 그때 그때마다 요구하는대로 입력만 해주면 자세한 소개를 해준다. 여기서는 따로 데이터베이스의 의미는 설명하지 않고 dBASE의 사용법을 소개하고자 한다.

# 디베이스의 구동

데이터베이스에 대해 어느정도 감이 잡혔다면 Tutoral에서 소개한 바와 같이 A: 드라이브에 시스템 #1을 넣고

```
A>dbase ↵
```

라고 입력하면 시스템 #2 디스켓의 삽입을 원하는 메시지가 나온다. 그러면 시스템 #1 디스켓을 빼고 시스템 #2 디스켓을 A: 드라이브에 넣고 리턴을 누르면 디베이스를 시작하게 된다. 이때 B:드라이브에는 데이터를 저장할 빈 디스켓을 넣어야 한다. 만약 하드디스크를 사용한다면 하드디스크에 따로 디렉토리를 만들어 그곳에 시스템 #1, #2를 모두 카피한 후 디베이스를 구동시키면 될 것이다.

이때 주의할 점은 디베이스의 초기화면이 처음부터 메뉴형식으로 디베이스를 사용할 수 있는 ASSISTANT 상태로 되어 있다면 〈Esc〉를 눌러 도트 프롬프트 상태로 갈 수 있지만 다음과 같은 작업을 거쳐 곧바로 도트 프롬프트가 나오도록 하길 바란다(이 강좌를 편리하게 설명하기 위함이며 어느정도 익숙해지면 도트 프롬프트에서 사용하는 것이 더 편리함을 느끼게 될 것이다. ASSISTANT는 디베이스 사용법을 전혀 모르는 사용자를 위한 것으로 명령어를 익히기 전까지는 편리하게 사용할 수 있으나 그 기능이 제한되어 있어 처음부터 도트 프롬프트 상태에서 익히는 것이 좋을 듯하다).

디베이스는 처음 구동이 되자마자 시스템 #1의 CONFIG. DB라는 텍스트 화일을 찾아 그 내용에 따른 환경을 설정한 다음 디베이스를 작동시킨다. 대부분 독자는 처음에 다음과 같은 내용의 CONFIG. DB화일을 갖고 있을 것이다. 이 화일의 내용은 아래와 같다.

```
status=on
command=assist
```

이의 내용은 22번 라인의 STATUS BAR를 나타나게 해주고 시동후 첫번째 명령은 assist라는 명령임을 알려주고 있다. 독자가 사용하는 에디터나 워드프로세서를 이용하여 command=assist라는 부분을 지우면 바로 도트 프롬프트 상태로 들어갈 수 있다. 그리고 하드디스크를 사용하지 않을 경우는 다음을 추가하길 바란다.

```
default=b:
```

이는 데이터를 어느 드라이브에서 사용할 것인지를 미리 지정하는 것이다. 단, 하드디스크를 사용하는 경우라면 이 부분은 대신 디베이스 디렉토리 외에 데이터가 들어갈 디렉토리를 새로 만든 다음 AUTOEXEC. BAT 화일에 다음과 같은 내용을 추가시킨후 데이터 디렉토리에서 디베이스를 부르면 된다. 디베이스 디렉토리를 DB라고 정했다면

```
PATH C:\; C:\DB
```

과 같이 하면 된다.

디베이스를 구동시키기 전에 또 한가지 확인해 주길 바란다. 독자가 사용하는 MS-DOS의 CONFIG.SYS라는 화일에(하드디스크로 부팅하는 경우는 루트 디렉토리에)

```
FILES=20
BUFFERS=15
```

라는 내용이 포함되어 있는가 확인해보고 없을 경우는 앞의 내용을 추가해야 한다. 이 화일이 없을 경우는 새로 만들어 넣는다. 이는 MS-DOS가 동시에 열 수 있는 화일의 갯수와 버퍼의 크기를 지정해 주는 정보이다. 만약 수정하였을 경우는 다시 부팅을 해야 한다(컴퓨터는 ROM BIOS에 의해 초기에 작동이 되며 DOS의 BIOS. SYS, DOS. SYS라는 감추어진 화일을 읽는다. 이때 CONFIG. SYS라는 화일이 있을 경우 이의 환경에 의해 작동되고 없을 경우는 디폴트로 환경을 만든다.). 다음에는 AUTOEXEC.BAT 화일이 있는가 확인하여 배치화일을 실행시킨다. 이와 같이 CONFIG. SYS는 부팅시 한번만 읽기 때문에 다시 부팅한다.

이제는 디베이스를 부팅하여 보자. 그러면 **그림 1**과 같은 초기화면이 나오고 화면에서의 지시대로 리턴키를 누르면 바로 도트 프롬프트 상태로 들어가게 된다.

# 디베이스의 종료

어느 프로그램을 테스트 해보기 위해

〔그림 1〕 dBASE Ⅲ PLUS의 초기화면

```
                    dBASE III PLUS  Version 1.1
                   This Software is Licensed to:
                        The Name of the User
                      The Name of the Company
                             0000001-01

        Copyright (c) Ashton-Tate 1985, 1986.  All Rights Reserved.
   dBASE, dBASE III PLUS and Ashton-Tate are trademarks of Ashton-Tate

   You may use the software and  printed materials in  the dBASE III
   PLUS package under the  terms of the  Software License Agreement;
   please  read it.   In summary,  Ashton-Tate grants you a paid-up,
   non-transferable,  personal license to use dBASE III  PLUS on one
   computer  work  station.   You  do  not become  the owner  of the
   package nor do  you have  the  right  to  copy  (except permitted
   backups  of  the  software)  or  alter  the  software  or printed
   materials.   You are legally accountable for any violation of the
   License Agreement and copyright, trademark, or trade secret law.

Command Line      <C:>

     Press ↵ to assent to the License Agreement and begin dBASE III PLUS.
```

구동시키기는 하였으나 어떻게 빠져 나가야 하는지 몰라서 결국엔 새로이 부팅을 해야하는 경우를 경험해본 적이 있다. 디베이스의 종료 명령은 QUIT를 입력함으로써 DOS로 빠져 나가게 된다. 이는 디베이스 명령중에 가장 중요한 명령이라 생각한다. 왜냐하면 QUIT로 빠져 나가지 않으면 데이터 화일에 손상을 입힐 우려가 있기 때문이다. 흔히 프로그램에서 DOS상태로 빠져 나가는 명령으로 다음과 같은 것들이 있으니 참고하길 바란다. 필자의 경험으로 크게 실수한 적이 있어서 소개하는 바이다.

**QUIT, EXIT, LOGOUT, LOGOFF, SYSTEM, 〈Ctrl-C〉, 〈ESC〉**

# 디폴트 드라이브의 변경

앞에서 하드디스크를 사용하지 않는 경우에 디폴트 드라이브를 B:로 지정하기 위해 CONFIG.DB에 그 내용을 명시했으나 하드디스크를 사용하는 경우도 데이터를 디스켓에 입력할 경우는 CONFIG.DB에 그 내용을 추가한다. 또한 필요시에는 디폴트 드라이브를 변경할 경우가 생긴다. 이때에는 다음과 같은 명령으로 간단히 해결할 수 있다.

```
. SET DEFAULT TO A:
```

*디폴트 드라이브를 A:로 정할 경우

이제부터는 화일 이름앞에 드라이브명을 정하지 않으면 A: 드라이브로 간주하고 화일을 찾거나 생성시킬 것이다.

# 편리한 기능들

## ●화면 소거

다음은 화면을 깨끗이 지우는 명령이다. 도스의 CLS와 같은 것으로 디베이스에서는 CLEAR라는 명령을 사용한다.

```
. CLEAR
```

*도스에서는 CLS가 CLear Screen이라는 것을 상기하자.

## ●계산 및 질문

디베이스에서는 도트 프롬프트 상태에서 직접 계산이 가능하다. 다음의 명령을 입력해 보자.

```
. ? 123+456
579
. ? 5+10-2*2/2
  13.00
. AAA=2
2
. BBB=15
15
. ? AAA-BBB
       -13
. ? BBB**AAA
      225.00
. ? BBB
        15
```

위와 같이 ?는 계산 결과를 물어보거나 변수의 내용을 물어 보는 명령이다. 다시 말해 보면 계산된 값이나 변수값을 화면에 프린트 해주는 명령이 된다. 변수명=〈데이터〉는 변수에 데이터를 저장해주는 명령이다. 위의 변수 지정 명령은 다음과 같은 방법으로도 가능하다.

```
. STORE 2 TO AAA
2
```

*AAA=2라는 명령과 동일하다.

```
. STORE 5 TO CCC,DDD,EEE
5
. ? AAA
         2
. ? CCC
         5
. ? DDD,EEE
         5       5
```

STORE TO를 사용하면 같은 내용의 데이터를 여러가지 변수에 한꺼번에 지정할 수도 있음은 위의 명령으로 알 수 있을 것이다.

## ●BOX 그리기

```
. @ 3,10 TO 19,60
. @ 5,20 TO 17,50 DOUBLE
```

이라는 명령을 입력해 보길 바란다. 무슨 명령인지 단번에 알게 될 것이다. 우리가 사용하는 모니터는 그래픽 모드에서는 비디오 카드에 따라 다르겠지만 텍스트 모드에서는 25행 80자 형태의 CELL로 되어 있다.

디베이스에서는 좌측상단 맨끝을 0, 0

* '*'는 여기서 편의상 주석임을 나타내기 위해서 사용하였다.

으로 지정해 놓고 우측하단 맨끝은 24, 79로 정하여 BOX를 그릴 경우는 이 명령에서 BOX의 좌측상단 끝점과 우측하단 끝점을 알려주면 된다. DOUBLE이라는 옵션을 주면 2줄의 BOX를 그려준다.

```
. @ 7,30 TO 7,40  *가로 줄을 그린다.
. @ 8,65 TO 20,65 DOUBLE *세로 줄을 그린다.
. @ 4,10 CLEAR TO 20,40 *BOX 크기만큼 화
                         면을 지운다.
```

## HISTORY와 간편한 TYPE

여기까지 입력해 보았다면 이번에는 화살표 키를 눌러 보길 바란다. 뭔가를 느끼게 될 것이다. 한번 사용한 명령은 디폴트로 20개까지 기억하고 있다가 다시 사용할 때는 위 화살표 키를 사용하면 간편하게 입력을 할 수 있다. 이 명령은 명령어를 자주 입력함으로써 빨리 익힐 수 있도록 하기 위하여 좀 더 나중에 설명하려 했으나 디베이스의 편리함을 알려주기 위해 먼저 설명한다.

```
. SET HISTORY OFF        *이제부터는 화살표
                          키로 Recall 할 수가 없다.
```

다시 화살표 키를 이용하여 Recall 기능을 되살리고자 하면

```
. SET HISTORY ON
```

을 입력하면 된다.

그러나 여기서 인간이 얼마나 간사한지 발견할 수 있다. 방금 SET HISTORY ON이라는 명령을 다시 치기 위해 타입을 하면서 '아! 불편하다. 화살표 키를 이용하면 SET HISTORY OFF에서 FF만 살짝 N으로 바꾸어서 리턴을 치면 되는데…'하는 생각을 하게 된다. 여기에 이러한 분을 위해 또 한가지 배려해준 것이 있다. 어느 프로그램 어느 랭귀지에서도 찾아 볼 수 없는 것이다. 디베이스의 모든 명령은 아무리 긴 것이라도 4자만 쳐도 디베이스는 이해하게 되어 있다.

그러므로 위의 SET HISTORY ON이라는 명령은 SET HIST ON이라는 명령으로 타입하여도 무방하다. 그러나 4자 이후의 것을 무시하면서 읽는 것이 아니므로 SET HISTOLY ON이라고 타입하면 에러를 발생하게 될 것이다. SET HISTO ON이라 해도 가능하므로 잘 모르면 4자 까지만 입력하여 에러를 방지할 수 있다(얼마나 좋은 기능인가, 이 기능을 보면 더욱더 디베이스를 사랑하게 된다. 그러나 요즘은 이 기능을 거의 사용하지 않게 되었다. 이글을 쓰면서도 4자까지만 입력하여 쓴다면 아마 무슨 내용인지 도저히 모르는 독자가 있을 것 같아서…).

## DOS 명령어

디베이스에서는 DOS에서 사용하는 일부 명령을 사용할 수 있다.

**그림 2**에서 DIR을 제외하고는 디베이스에서는 와일드카드를 사용할 수 없고, 디베이스 명령중 DOS 명령을 직접 사용하는 명령으로는 RUN/ ! 이라는 명령을 사용할 수 있다.

```
. RUN DIR
. ! COPY AAA.PRG BBB.PRG
. ! DEL *.BAK
```

## 화면정보

다음의 명령을 차례로 타이프 해보길 바란다.

```
. SET STATUS OFF
. SET SCOREBOARD OFF
. SET STATUS ON
. SET SCOREBOARD ON
```

〔그림 2〕 DOS와 디베이스의 명령 비교

| DOS | | 디베이스 |
|---|---|---|
| A> TYPE AAA.PRG | ⟶ | . TYPE AAA.PRG |
| A> COPY AAA.PRG BBB.PRG | ⟶ | . COPY FILE AAA.PRG TO BBB.PRG |
| A> RENAME BBB.PRG CCC.PRG | ⟶ | . RENAME BBB.PRG TO CCC.PRG |
| A> DEL CCC.PRG | ⟶ | . DELETE FILE CCC.PRG |
| A> ERASE CCC.PRG | ⟶ | . ERASE CCC.PRG |
| A> DIR/W | ⟶ | . DIR *.* |

```
. SET MESSAGE TO "ABCDEFG"
. SET MESSAGE TO
```

SET STATUS ON/OFF : 22번 라인의 상태 표시선 표시 여부를 결정한다.

SET SCOREBOARD ON/OFF : SET STATUS OFF인 경우에는 0번 라인에, SET STATUS ON인 경우에는 STATUS BAR에 상태표시를 해준다.

SET MESSAGE TO "……" : SET STATUS ON인 경우 화면 맨 밑줄에 표시되는 Enter a dBASE Ⅲ PLUS Command. 라는 내용을 사용자가 마음대로 바꾸고자 할 경우에 사용하며 SET MESSAGE TO 〈Enter〉라고만 입력하면 디폴트 상태인 Enter a dBASE Ⅲ PLUS Command. 라는 내용이 나온다.

## 메모리 변수의 종류

디베이스에서는 외부에서 데이터를 저장하는 특별한 형태의 변수인 메모리 변수가 있는데 이 변수들은 대개 데이터의 임시 기억장소로 사용된다. 예를 들면 어떤 필드의 계산 결과를 저장했다가 다음의 계산에 그 변수와 내용을 사용할 수 있다.

이들 변수는 Character, Numeric, Date, Logical의 4가지가 있다.

1. Character변수는 문자열을 기억하는 변수이다.

```
. name="Bahk Seonghyeon"
. count="1000"
. birth="04/25/1961"
```

위의 변수는 모두 문자형 변수로 기억되며 최대 254문자까지 가능한데 실

제 메모리는 문자수보다 2바이트 더 차지한다.

2. Numeric 변수는 계산이 가능한 수치로 기억된다. 위의 count라는 변수의 경우는 숫자를 기억하고 있지만 계산이 가능한 수치값이 아니라 글자로서의 수인 것이다. 이 변수는 정확도가 15.9자리인데 보통 15자리 정도가 정확성을 가지고 0이 아닌 숫자를 비교하는 데는 13자리 정도이다.

```
. amount=30000
. result=100-200+400
     300
. tot=amount-result
   29700
```

위의 변수는 모두 수치인 변수이다.

3. Date 변수는 날짜를 기억하는 변수이다. 날짜와 날짜간의 계산이 가능하도록 하기 위해 날짜 변수를 이용한다. 날짜 변수는 함수를 사용하여야 변수에 입력이 가능하다.

DATE( )는 현재의 날짜를 가르쳐 주는 함수이고 CTOD( )는 문자형 변수를 날짜형 변수로 바꾸어 주는 함수이다.

```
. today=DATE()
10/04/88
. birthday=CTOD(birth)
04/25/61
. ? birth-today
   10023
. ? today-100
06/25/88
. ? today+100
01/11/89
```

4. Logical 변수는 True와 False의 구별을 위한 변수로서 참과 거짓을 대변하는 한 글자만을 받아들인다(T, t, Y, y, F, f, N, n으로 입력할 수 있다).

```
. STORE today=birthday TO answer
. ? answer
.F.
. check=.y.
.T.
```

메모리 변수는 256개의 변수를 6000 바이트까지 사용할 수 있으며 CONFIG.DB에 의해 더 늘릴 수도 있다.

또 각 형태의 메모리 변수는 **그림 3**과 같이 서로 다른 형태의 변수로 바꿀 수도 있다.

단, 변수명이 디베이스 명령어와 같아지지 않도록 주의해야 한다.

이번호에서는 디베이스를 소개하는데 그 중점을 두었다. 실제 데이터베이스 화일은 아직 작성해 보지도 않았을 뿐더러 간단한 몇가지 명령만 설명해 보았다. 디베이스의 본격적인 강좌는 다음호부터 시작할 것이다. 우선 디베이스의 세계에 들어오게 된 독자를 축하하는 바이다.

〔그림 3〕 변수형태의 변환 예

| | | | |
|---|---|---|---|
| Character → Numeric | VAL 함수 이용 | 예) VAL(count) |
| Character → Date | CTOD 〃 | 예) CTOD(birth) |
| Numeric → Character | STR 〃 | 예) STR(amount) |
| Date → Character | DTOC 〃 | 예) DTOC(today) |

이번호에 소개된 명령어―매뉴얼을 찾아 보면서 복습을 한다면 더욱 많이 알게 될 것이다. 이해가 가지 않는 부분이 있을 것이나 차후에 설명할 예정이므로 한번 정도 읽어 보는 것이 바람직할 것이다.

```
QUIT
SET DEFAULT TO
CLEAR
?
STORE
SET HISTORY ON/off
@ yy,xx TO yy,xx [DOUBLE]
@ yy,xx CLEAR TO yy,xx
TYPE
COPY FILE
RENAME
DELETE FILE / ERASE
DIR
RUN / !
SET STATUS ON/off
SET SCOREBOARD ON/off
SET MESGAGE TO
DATE()
CTOD(C)
DTOC(D)
VAL(C)
STR(N)
```

0; boot sector, 512 Byte
1~4 ; FAT | 5~11 Directory | 12~; Data
IBMIO.COM
IBMDOS.COM
1 cluster = 2 sector



# 디스크 논리구조를 이용한 화일복구

플로피디스크와 하드디스크는 컴퓨터 사용자들에게 가장 널리 알려져 있는 컴퓨터 보조기억장치이다. 게다가 기억용량 또한 갈수록 늘어나고 있어서 그 인기는 좀처럼 흔들리지 않고 있다. 지금 이순간에도 컴퓨터를 다루는 독자들 옆에는 언제나 디스켓이 따라 다니고 있을 것이다. 손에, 가방속에 …마치 그림자처럼.

워드프로세서로 작성한 각종 데이터들, 또는 응용프로그램들을 손쉽게, 단 한순간의 조작으로 디스크속에 기억시킬 수 있고 언제든지 다시 꺼낼 수가 있으므로 속도와 편리함은 이루 말할 수 없다. 그렇지만 바로 이런 이유로 물, 공기의 은혜를 쉽게 잊어 버리듯 자칫 디스크의 내부사정에는 무관심할 때가 많다. 효과적인 화일운용과 어셈블리작업을 위해서는 우선 디스크의 내부구조를 이해하는 것이 급선무라 하겠다. 사실, 디스크속에서 이루어지는 놀라울 정도의 섬세한 비트작업들은 독자들을 매료시키기에 충분하리라 본다.

그리고 하드디스크는 플로피디스크에 비해 엄청난 용량을 가지므로 약간의 기술적인 차이는 있으나, 근본구조에는 동일하기 때문에 여기서는 독자들이 쉽게 접할 수 있는 플로피디스크를 중심으로 설명하고자 한다.

## 디스크의 논리구조

플로피디스크도 크기와 기록방식에 따라 여러가지로 나누어지지만 근본구조는 같다고 할 수 있다. 그래서 여기서는 DOS 2.0 이후의 IBM PC/XT 또는 그 호환기종에서 널리 사용되는 배정도 배면디스켓(Double Density Double Side Diskette)을 예로 들었다.

디스크를 처음 구입하게 되면 DOS 유틸리티 프로그램 FORMAT.COM을 이용하여 포매팅시켜야 하는데 이것은 데이터를 읽고 쓸 수 있게 디스크를 초기화하는 작업을 수행한다. 이때 이루어지는 작업내용을 **그림 1**에 나타냈다. 그림 1에서 알 수 있듯이 포매팅 후 디스크의 전체용량은

40Track＊9Sector＊2Side＊512Byte=360Kbyte

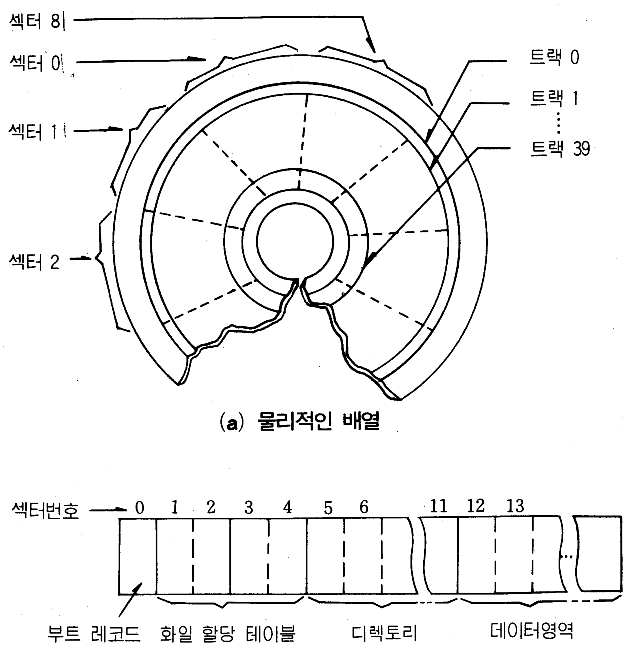
가 된다.

그리고 0섹터, 즉 디스크에서 최초로 시작되는 섹터에는 부트레코드(Boot-Record)가 저장되고 이어지는 1, 2섹터와 3, 4섹터에는 화일할당테이블(FAT : File Alloction Table) 영역이, 그 뒤로 5, …, 11섹터까지는 디렉토리영역(Directory)으로 정의된다. 그리고 12섹터부터는 데이터영역이 되는데 만약 포매팅작업시 '/S' 옵션을 이용하면(즉, DOS시스템을 포함할 때) IBMIO.COM과 IBMDOS.COM이 여기부터 들어가게 된다.

### (1) 부트 레코드

여기서는 IBMIO.COM과 IBMDOS.COM(기종에 따라 명칭이 다를 수도 있음)을 부트시켜주는 프로그램 루틴이 들어 있다. 즉, 컴퓨터시스템의 전원스위치를 ON시키면 롬(ROM)속에 들어 있는 롬바이오스(ROMBIOS) 프로그램이 실행되어 A드라이브(만약 드라이브가 비어 있으면 하드

〔그림 1〕 디스크 포맷




(a) 물리적인 배열

(b) 논리적인 배열

디스크)의 부트레코드로 제어를 넘겨준다. 제어를 넘겨받은 부트레코드 프로그램은 무조건 IBMIO. COM과 IBMDOS. COM프로그램을 읽으려고 하는데, 만약 지정위치(12 섹터)에 이들 프로그램이 없으면 'Non system disk or Disk error'라는 메시지를 출력한다. 아마 독자들도 몇번은 이런 경우를 경험했으리라 짐작한다. 그리고 즉시 '이것은 DOS 시스템 프로그램이 없는 디스크로구나'라고 판단한 후 올바른 DOS 디스크로 다시 부팅시켰을 것이다.

## (2) 화일 할당테이블

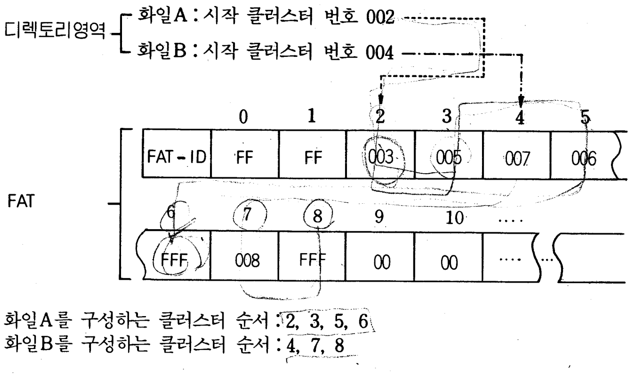
디스크에서 화일들은 어떤 상태로 기억되고 있을까? DOS는 각 화일들을 클러스터(Cluster)(*)라는 단위로 쪼개어 작업을 하기 때문에 디스크내에서는 일정한 크기의 클러스터들이 여기저기 흩어져 있게 된다. 그러나 여러 화일들을 이루고 있는 클러스터들이 무질서하게 흩어져 있는것은 아니다. 혹은 이어져 있고 혹은 떼어져 있는 상태로 있지만, 이들은 보이지 않는 줄로 서로 연결되고 있다. 바로 이 줄을 만들어 주고 있는 부분이 FAT(File Allocation Table) 영역이다. (그림 2 참조).

처음에는 동일한 화일을 구성하는 클러스터들이 순서적으로 배열되어져 액세스효율도 좋고 보기에도(?) 좋은 상태로 된다(물론 외관으로 디스크내의 클러스터 배열상태를 볼 수는 없지만 컴퓨터시스템을 자주 접하는 유저들은 각종PC툴(**) 등을 갖고 있어서 언제든지 디스크속으로 들어가 그 배열상태를 볼 수 있다.

그후, 계속 작업을 하나보면 불필요해진 화일을 지우기도 하고 또 새로운 화일을 첨가하기도 하면서 클러스터 배열은 점차 무질서하게 보이기 시작한다. 즉, 중간 위치의 화일을 지우고 새로운 내용을 첨가할 때 저장하다가 남은 부분은 빈 클러스터를 계속 찾아가며 저장되기 때문에 그렇다. 그래도 화일은 제대로 액세스되며 다만 흩어진 정도가 클수록 액세스효율이 떨어질 뿐이다.

그림 2를 보면 클러스터들의 순서가 어떻게 형성되고 있는지를 도식으로 쉽게 설명하고 있다. 우선 디렉토리영역에서 화일A의 첫 클러스터를 읽은 후 다음에 이어지는 클러스터를 찾기 위해 FAT영역을 참조하게 된다. 즉 002번에 할당된 FAT에는 003번이 지정되어 있고, 다시 003번 할당영역을 참조하면 005가 지정되어 있다. 이렇게 계속해서 006번 할당영역까지 가 보면 'FFF'란 수치를 만난다. 이 'FFF'란 수치는 화일 A의 클러스터가 여기서 끝남을 의미하고

〔그림 2〕 도식으로 표현한 FAT




있다. 그래서 화일 A의 FAT내에서의 클러스터는 2,3, 5, 6이 되며 화일 B도 같은 방법으로 순서를 찾게 된다.

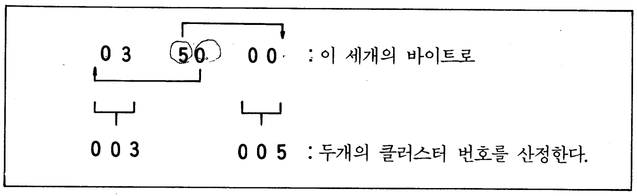
다음은 FAT에서 짚고 넘어가야 할 몇가지 사항을 간추려 본 것이다.

① 실제 FAT에서는 3바이트를 가지고 2개의 클러스터번호를 지정하고 있다. 예를 들면 그림 2에서 003과 005두개의 클러스터값은 실제로 '03 50 00'라고 표시되어 있다. 그림 3에 이의 산정방법을 보였다.

② FAT의 첫 바이트는 디스크 종류를 판별하는 지시자(Identifier)로 쓰이게 된다. 그런데 여기서는 3바이트씩 연관되어 클러스터값을 산정해야 하므로 어쩔 수 없어 처음 부분의 3바이트는 못쓰게 되어 버린다. 그래서 000번과 001번을 지나쳐 002번이 최초의 클러스터값이 되고, 이 'FAT ID, 000,001번에는 항상 'FD FF FF'라는 수치로 채워진다(화면 1 참조).

③ 이 FAT에서 설정된 값들은 단지 클러스터들의 올바른 순서만을 나타내고 있으므로 이에 해당하는 실제 데이터를 액세스하기 위해서는 섹터의 실제 상대위치를 구해야 한다(식 1 참조).

〔그림 3〕 클러스터번호 산정방법



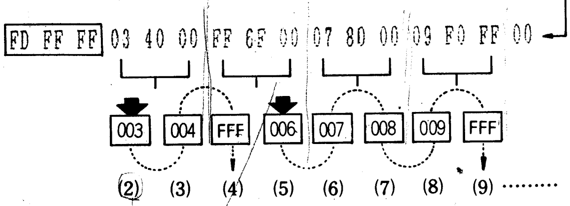

* 원래 클러스터라는 말은 어떤 개체들이 한데 뭉쳐진 양태를 나타내는데 여기서는 2개의 섹터모임을 의미하고 있다. 그러나 모든 디스크시스템에서 이 값을 갖고 있는 것은 아니며, 이 경우(1Cluster=2Sector)는 360Kbyte 용량의 배면배밀도 디스크인 경우이다. 일반적으로 클러스터는 섹터의 2의 멱승값을 취한다.

**노턴유틸리티나 Q-DOS, PCTOOLS 등이 그것이다. 본고에서도 각종 덤프리스트들은 PC TOOLS DELUXE R4.11로 작업했고, 어셈블리 프로그래밍작업에서는 특히 DEBUG 유틸리티가 훌륭한 역할을 해 주었다.

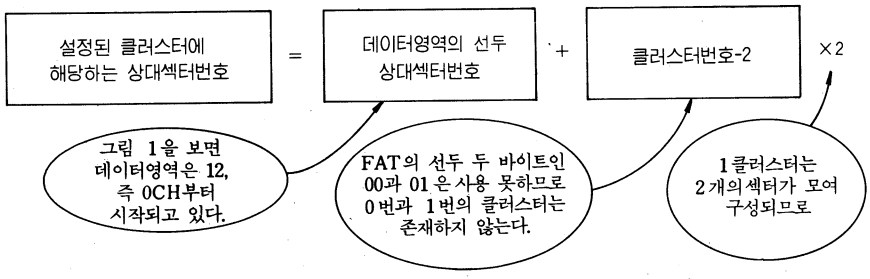
〔화면 1〕 FAT 덤프리스트 예

```
Path=A:
                    Absolute sector 00001, System FAT

Displacement ----------------- Hex codes-------------------  ASCII value
 0000(0000) FD FF FF 03 40 00 FF 6F 00 07 80 00 09 F0 FF 00     @  o
 0016(0010) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0032(0020) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0048(0030) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0064(0040) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0160(00A0) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0176(00B0) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0192(00C0) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0208(00D0) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0224(00E0) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0240(00F0) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```




〔식 1〕 실제 데이터를 액세스하기 위해 상대섹터를 구하는 식




〔화면 2〕 디렉토리영역 덤프리스트 예

화일 이름 / 화일크기 / 속성

```
                    Absolute sector 00005, System ROOT

Displacement ----------------- Hex codes-------------------  ASCII value
 0000(0000) 55 4E 44 45 4C 20 20 20 43 4F 4D 20 00 00 00 00  UNDEL   COM
 0016(0010) 00 00 00 00 00 00 EC 04 21 00 02 00 13 08 00 00          !
 0032(0020) 55 4E 44 45 4C 20 20 20 42 41 53 20 00 00 00 00  UNDEL   BAS
 0048(0030)                      77 02 21 00 05 00 B5 12 00 00      w !
 0064(0040) 00 00 00 00 00             58 54 20 00 00 00 00  HKFILE TXT
 0176(00B0) 00 00 00 00 00 00 00          00 D6 00 00 00 00      H !
 0192(00C0) 00 00 00 00 00 00 00 00 00          00 00 00
 0208(00D0) 00 00 00 00 00 00 00 00 00 00 00 00          00
 0224(00E0) 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0240(00F0) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

### (3) 디렉토리

FAT영역의 바로 다음에는 등록화일에 대한 각종 정보가 들어있는 디렉토리가 위치하고 있다. 그림 1에서 나타냈듯이 5섹터에서 11섹터까지의 7섹터가 할당된다. 즉, 그 크기는

7Sector＊512Byte＝3584Byte

가 된다. 그리고 여기서는 하나의 화일을 지정하는데, 32바이트가 할당되며 지정가능한 화일의 최대수는 112개가 된다.

3584/32＝112

화면 2와 그림 4에 디렉토리 영역을 나타내었는데, 화일 정보들이 어떻게 배열되어 있으며, 또 어떤 형태로 기억되고 있는지를 살펴보기 바란다. 특히 화일이름의 첫 문자와 시작 클러스터번호, 그리고 화일 크기는 화일을 복구할 때 중요한 정보가 된다.

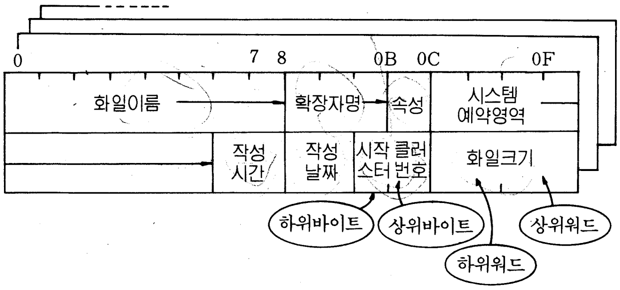
- 디스크에서 어떤 화일을 지우면 DOS는 그 화일의 첫 문자를 'E5H'로 바꾸고, FAT에서 그 화일에 해당하는 연결선(Link)을 모두 '00'로 채운다.
- 그러나 다행히도 디렉토리에 시작 클러스터번호와 화일 크기는 그대로 남아 있어서 이런 잔여정보를 이용해서 본래의 화일로의 복구가 가능한 것이다.

화일을 삭제하면 FAT내의 연결선(링크)이 사라지고 디렉토리의 해당화일 첫 문자가 'E5H'로 바뀌는 것 외에 별다른 변화는 없으므로 데이터들은 그대로 보존된 상태이다. 다시 말해 이 변화된 부분을 원래대로 회복시킴으로서 지워졌던 화일을 재생시킬 수가 있는 것이다.

## PSP

DOS기능을 이용해서 화일을 액세스하고 이 원리를 가지고 실수로 지워진 화일을 복구하는데, 이 PSP(Program Segment Prefix), 특히 FCB에서의 정

〔그림 4〕 디렉토리영역의 구조




보응용은 상당히 중요하게 작용되어질 수 있다.

## (1) PSP란?

즉시 실행가능한 화일(여기서는 .COM 화일을 중심으로 한다)을 주기억장치로 로드하면 해당 세그먼트(코드세그먼트) 내의 앞부분 100H에는 PSP라는 부분이 DOS에 의해 자동적으로 들어서게 된다. 그 구성은 아래와 같다.

- 환경부분(Environment)…패스명 등 환경정보가 있다.
- FCB(File Control Block)…화일조작, 전송 등에 필요한 각종 정보가 들어가 있으며, 화일조작은 여기에 수록되어 있는 정보를 이용함으로써 가능하게 된다.
- DTA(Data Transfer Area)…디스크에서 화일을 입출력할 때 쓰이는 버퍼이다.

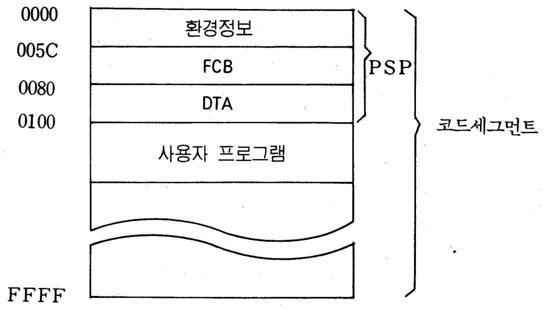
이 영역 때문에 COM화일을 작성할 때에는 'ORG 100H' 명령에 의해 미리 PSP영역 다음의 100H번지부터 실행될 수 있는 상태로 기억된다. 그리고 모든 레지스터의 PSP의 선두위치를 가리키게 된다. 그림 5는 주기억장치에 프로그램이 로드된 상태를 보인 것이다.

## (2) FCB와 DTA

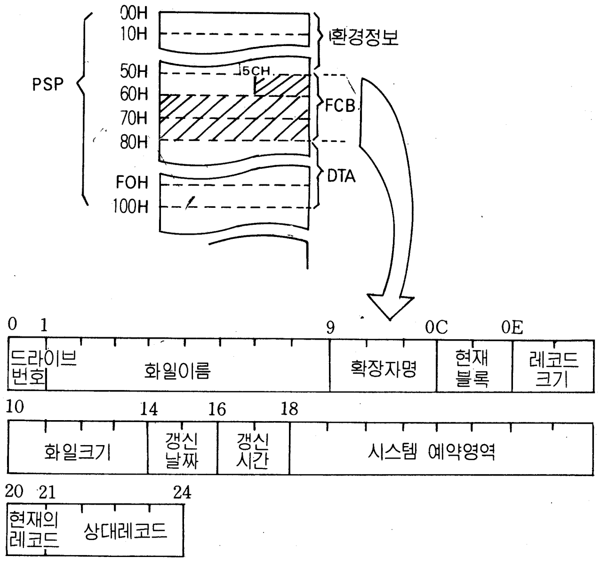
FCB는 5CH번지부터 시작되며 화일을 열거나 닫는 역할 및 입출력에 필요한 정보들이 있다. **그림 6**과 **화면 3**에 그 형식과 실제 예를 보였다. 바로 이 부분이 삭제된 화일을 복구시키는데 가장 중요한 단서가 되는 곳이므로 잘 익혀두기 바란다. 또 한가지 여기서 FCB가 두 부분으로 나누어 있는 것은 CP/M과의 호환성 때문이므로 여기에서는 무시해도 좋다.

다음 DTA(Data Transfer Area)는 데이터 전송영역으로써 메모리상의 특정영역과 디스크를 중계하는 역할을 담당한다. 즉, 모든 데이터전송을 할 때 반드시 이 DTA에 보관되었다가 실제위치에 쓰여지거나 또는 로드된다. 이 영역은 유저가 별도로 지정하지 않는 한 그림 5에서 보듯이 PSP의 80H-FFH 범위로 지정된다.

〔그림 5〕 COM화일이 로드된 메모리




〔그림 6〕 PSP와 FCB, DTA의 배열관계




(주) *드라이브 번호 : 0이면 현재의 드라이브, 1이면 A:, 2이면 B:
*레코드 크기 : FCB의 6A번부터 수록되며 별도 지정이 없으면 80H(128 바이트)가 지정된다.
6AH…레코드 크기의 하위 바이트
6BH…레코드 크기의 상위 바이트
*화일 크기 : 6CH번지, 6DH번지 ― 화일 크기의 하위 워드
6EH번지, 6FH번지 ― 화일 크기의 상위 워드

# 프로그램 설명

**리스트 1**과 **리스트 2**는 지워진 화일에 남아 있는 정보들을 이용해서 그 화일을 복구시키는 프로그램이다. 리스트 2의 베이직프로그램은 디렉토리에 명시되어 있는 시작 클러스터만을 참조하기 때문에 1클러스터, 즉 1K바이트분을 회복시킨다. 만약 1K바이트 이상의 크기를 갖는 화일인 경우 1K바이트 부분에서 짤려진 채 복구되는 것을 볼 수 있다(**화면 4** 참조).

리스트 1의 어셈블리어 프로그램은 디렉토리를 참조하여

〔화면 3〕 FCB 덤프리스트 예

```
A:\>dir

 Volume in drive A has no label
 Directory of  A:\

UNDEL    COM     2077   1-01-80   12:15a
COUNT    ASM     5965  10-16-85    4:00a
DEBUG    COM    15897   3-17-87   12:00p
        3 File(s)    336896 bytes free

A:\>debug
-a100
1045:0100 mov dx,5c
1045:0103 mov ah,f
1045:0105 int 21
1045:0107 int 20
1045:0109
-ncount.asm  ←─── (OPEN하고자 하는 화일이름)
-d0
1045:0000  CD 20 00 A0 00 9A EE FE-1D F0 F4 02 96 0C 2F 03
1045:0010  96 0C BC 02 96 0C 76 0B-01 01 01 00 02 FF FF FF
1045:0020  FF FF FF FF FF FF FF FF-FF FF FF FF 59 0C 4E 01
1045:0030  D0 0F 14 00 18 00 45 10-FF FF FF FF 00 00 00 00
1045:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
1045:0050  CD 21 CB 00 00 00 00 00-00 00 00 00 00 43 4F 55
1045:0060  4E 54 20 20 20 41 53 4D-00 00 00 00 00 20 20 20   ←── (FCB)
1045:0070  20 20 20 20 20 20 20 20-00 00 00 00 00 00 00 00
```

첫 문자가 'E5H'인 화일을 우선 찾는다. 그리고 사용자가 지정한 화일이름(지워진 화일이름)과 첫 문자만 제외하고는 꼭 같은 화일인지를 조사한다. 또 시작 클러스터번호와 화일 크기를 참조하여 FAT영역의 모든 공백엔트리들을 조사하며 연결자를 만들어 간다. 도중, 이미 다른 화일로 겹쳐 써졌으면 에러 메시지를 출력하고 작업을 끝내 버린다. 실수로 중요한 화일이 지워졌더라도 당황하지 말고(데이터는 여전히 남아 있으므로) 느긋하게 적절한 처리법을 구상하는게 중요하다. 먼저 어셈블, 링크시켜서 얻은 UNDEL.EXE 화일을 UNDEL. COM 화일로 변환(EXE2BIN. EXE 이용)한 다음 DOS 프롬프트상에서

'UNDEL〈화일이름·확장자명〉 '

을 입력한다. 드라이브명 지정도 가능하며 프로그램을 실행시켰을 때 화일이 지워졌던 당시 그대로 디스크가 보존되어 있으면 완전한 화일복구가 이루어진다(화면 5 참조). 그렇지만 동시에 여러 화일들의 삭제나 새로운 내용의 복사시는 완전한 복구가 힘들다. PC

〔화면 4〕 리스트 2 프로그램을 이용한 화일복구

```
A:\>DIR

 Volume in drive A has no label
 Directory of  A:\

UNDEL    COM     2067   1-01-80   12:20a
COUNT    ASM     5965  10-16-85    4:00a
        2 File(s)    353280 bytes free

A:\>DEL COUNT.ASM

A:\>
```

```
           Absolute sector 00001, System FAT

Displacement ----------------- Hex codes--------------------
 0000(0000)  FD FF FF 03 40 00 FF 6F 00 07 80 00 09 A0 00 FF
 0016(0010)  0F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0032(0020)  00 00 00 00 00 00 00 00 00 00
 0048(0030)  00 00 00 00 00 00 00
 0064(0040)  00 00 00 00
```

(a) 원래의 FAT

```
           Absolute sector 00001, System FAT

Displacement ----------------- Hex codes--------------------
 0000(0000)  FD FF FF 03 40 00 FF 0F 00 00 00 00 00 00 00 00
 0016(0010)  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0032(0020)  00 00 00 00 00 00 00 00 00 00 00
 0048(0030)  00 00 00 00 00 00 00 00 00 00
 0064(0040)  00 00 00 00 00 00
```

(b) COUNT.ASM 삭제후의 FAT

```
A:\>UNDEL COUNT.ASM

A:\>DIR

 Volume in drive A has no label
 Directory of  A:\

UNDEL    COM     2067   1-01-80  12:20a
COUNT    ASM     5965  10-16-85   4:00a
        2 File(s)    359424 bytes free

A:\>
```

```
Displacement ----------------- Hex codes--------------------
 0000(0000)  FD FF FF 03 40 00 FF 0F 00 00 00 00 00 00 00 00
 0016(0010)  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0032(0020)  00 00 00 00 00 00 00 00 00 00 00
 0048(0030)  00 00 00 00 00 00 00 00
  064(0040)  00 00 00 00 00
             00 00
```

(c) 복구후의 FAT (FAT는 복구 못하고 디렉토리의 첫 문자만 복구됐음)

〔화면 5〕 리스트 1 프로그램을 이용한 완전한 화일복구

```
A:\>DIR

 Volume in drive A has no label
 Directory of  A:\

UNDEL    COM     2077   1-01-80  12:15a
COUNT    ASM     5965  10-16-85   4:00a
        2 File(s)    353280 bytes free

A:\>DEL COUNT.ASM

A:\>UNDEL COUNT.ASM

A:\>DIR

 Volume in drive A has no label
 Directory of  A:\

UNDEL    COM     2077   1-01-80  12:15a
COUNT    ASM     5965  10-16-85   4:00a
        2 File(s)    353280 bytes free

A:\>
```

```
Displacement ----------------- Hex codes--------------------
 0000(0000)  FD FF FF 03 40 00 FF 6F 00 07 80 00 09 A0 00 FF
 0016(0010)  0F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0032(0020)  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0048(0030)  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0064(00                00 00 00 00 00 00 00 00 00 00 00 00
 00                           00 00 00 00 00 00 00 00 00
                                    00 00 00 00 00
```

(a) 원래의 FAT

```
Displacement ----------------- Hex codes--------------------
 0000(0000)  FD FF FF 03 40 00 FF 0F 00 00 00 00 00 00 00 00
 0016(0010)  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0032(0020)  00 00 00 00 00 00 00 00 00 00 00
 0048(0030)  00 00 00 00 00 00 00 00
  64(0040)   00 00 00 00 00
             00 00
```

(b) COUNT.ASM 삭제후의 FAT

```
Displacement ----------------- Hex codes--------------------
 0000(0000)  FD FF FF 03 40 00 FF 6F 00 07 80 00 09 A0 00 FF
 0016(0010)  0F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0032(0020)  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0048(0030)  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0064(00                   00 00 00 00 00 00 00 00 00 00 00
 008                          00 00 00 00 00 00 00 00 00
                                    00 00 00 00 00 00
```

(c) 복구후의 FAT (삭제하기 전과 동일하다)

〔리스트 1〕 UNDEL.ASM

```
        ;****************************************************
        ;이프로그램은 지워진 화일을 복구시키기 위한 것입니다.
        ;입력형식은 UNDEL <filename.ext>
        ;또는 UNDEL <드라이브명:filename>
        ;****************************************************

        FCB_LOC  EQU      5CH      ;FCB위치 설정
        SWITCH_LOC EQU    81H
        BIG_A    EQU      412FH
        SMALL_A  EQU      612FH
        CR       EQU      13
        LF       EQU      10

CODE_SEG         SEGMENT
        ASSUME   CS:CODE_SEG, DS:CODE_SEG, ES:CODE_SEG
        ORG      100H              ;.COM화일을 만들기
ENTRY:
        JMP   FIRST
        COPY_RIGHT          DB    '(C) 1988 Personal Computer'
        I_O_FLAG            DW ?   ;클러스터의 Read 혹은 Write 설정
        START_CLUSTER       DW ?   ;지워진 화일의 시작클러스터
        FILE_SIZE           DW ?   ;지워진 화일이 들어있는 드라이브
        DISK_DRIVE          DB 0
        NOT_FOUND_MSG       DB 13,'File not founded deleted$'
      WRITTEN_OVER_MSG   DB 13,'File already written over$'
      IS_IT_MSG          DB 13,10,'Is part of your file here? (Y/N) $'
      DATA          DB 1024 DUP(0)
      PROMPT_SECTOR      DB 512 DUP(0)

FIRST:                             ;시작

UNDEL      PROC     NEAR         ┐
           MOV      AL,CS:FCB_LOC │
           SUB      AL,1          │
           JNC      DRIVE_KNOWN   ├ 드라이브명 결정
           MOV      AH,19H        │
           INT      21H           ┘
DRIVE_KNOWN:
           MOV      DISK_DRIVE,AL
           MOV      DX,3             ;섹터 No.5부터 디렉토리영역
LOOP:
           ADD      DX,2             ;정확한 위치를 찾기위해 +2
           CMP      DX,11            ;디렉토리영역의 끝 ?
           JB       READ_DIR         ;아니오,점프
           MOV      AH,9             ;예,메시지를 출력하고 종료
           LEA      DX,NOT_FOUND_MSG
           INT      21H
           JMP      OUT              ;작업종료
READ_DIR:
           AND      I_O_FLAG,0       ;읽기선택
           CALL     CLUSTER_I_O      ;2섹터를 데이터영역으로 이동
           LEA      DI,DATA          ;지워진 화일을 찾기위해 준비
           MOV      AL,0E5H          ;지워진 화일의 첫문자는 'E5H'
           MOV      CX,400H          ;카운터를 1024바이트 설정
SEARCH:
REPNE      SCASB
           JCXZ     LOOP             ;계속 비교
           MOV      SI,FCB_LOC + 2   ;첫문자가 일치하면 다음문자 비교
           MOV      BX,11            ;화일명의 모든 문자 체크
CMPLOOP:
           DEC      BX
           CMPSB                     ;지워진 화일명과 비교
           JZ       CMPLOOP
           CMP      BX,0             ;모든 문자가 일치?
           JNZ      SEARCH           ;아니오, 체크계속
           MOV      AX,CS:FCB_LOC+1  ;예,첫문자 선택
           MOV      [DI-12],AX       ;첫문자를 지워진 위치로 이동
           OR       I_O_FLAG,1       ;쓰기 선택
           CALL     CLUSTER_I_O      ;복구된 화일명 등록
           MOV      AX,[DI+14]       ;FAT를 위해 시작클러스터지정
           MOV      START_CLUSTER,AX ;저장
           MOV      AX,[DI+16]       ;화일사이즈의 하위워드
           TEST     AX,1023          ;클러스터번호 찾음, MOD(size,1024)=0?
           JZ       EVEN_K           ;맞으면 그대로
           ADD      AX,1024          ;틀리면 다음 클러스터를 취함
EVEN_K:
           MOV      CL,10
           SHR      AX,CL
           MOV      DX,[DI+18]       ;화일사이즈의 상위워드
           MOV      CL,6             ;65536으로 곱하고,1024로 나눔
           SHL      DX,CL
           ADD      AX,DX            ;상위워드클러스터를 하위워드에 +
           MOV      FILE_SIZE,AX     ;화일사이즈에 저장
           MOV      DX,1             ;FAT영역 읽기
           AND      I_O_FLAG,0
           CALL     CLUSTER_I_O
           MOV      CX,FILE_SIZE
           MOV      AX,START_CLUSTER ;이미 다른 내용이 겹쳐졌는지 체크
           DEC      AX
           CALL     GET_NEXT_ZERO    ;다음 공백이 시작클러스터인가 판단
           CMP      DX,START_CLUSTER
           JE       FILL             ;예, FAT채움
           MOV      AH,13H           ;아니오,화일 겹쳐짐
           MOV      DX,FCB_LOC       ;기억시킨 엔트리를 지우고
           INT      21H
           MOV      AH,9             ;에러메시지를 출력
           LEA      DX,WRITTEN_OVER_MSG
           INT      21H
           JMP      OUT              ;작업 종료
FILL:                                ;FAT영역 채움
           MOV      AX,DX
           MOV      DX,0FFFH         ;엔트리의 끝지점(FFFH)
```

```
        CMP     CX,1                ;엔트리의 끝을 판단
        JZ      LAST                ;예,다음 엔트리를 더이상 체크할 필요없음
        CALL    GET_NEXT_ZERO
LAST:
        CALL    PUT_FAT_ENTRY
        LOOP    FILL                ;다음 클러스터를 조사
        MOV     DX,1                ;새로운 FAT를 위해 복사
        OR      I_O_FLAG,1          ;쓰기 선택
        CALL    CLUSTER_I_O
        MOV     DX,3                ;두번째 복사
        CALL    CLUSTER_I_O
OUT:
        INT     20H

UNDEL   ENDP

CLUSTER_I_O     PROC    NEAR        ;지정된 클러스터조사
  COMMENT* Put start sector in DX,loads Cluster into 'DATA' area*
        PUSH    AX                  ;사용된 레지스터들을 기억
        PUSH    BX
        PUSH    CX
        PUSH    DX
        PUSH    DI
        MOV     AL,DISK_DRIVE       ;드라이버 설정
        MOV     CX,2                ;1클러스터를 읽기위해 2섹터 요구
        LEA     BX,DATA             ;데이터영역 지정
        TEST    I_O_FLAG,1
        JNZ     WRITE
        INT     25H
        JMP     POPOUT
WRITE:
        INT     26H

POPOUT: POPF
        POP     DI                  사용했던 레지스터값 구함
        POP     DX
        POP     CX
        POP     BX
        POP     AX
        RET                         ;리턴
CLUSTER_I_O     ENDP

GET_NEXT_ZERO   PROC    NEAR
  COMMENT* FAT entry number(cluster #) in AX, returns next zero
        in DX*
        PUSH    AX
        PUSH    BX
        PUSH    CX
CHECK_NEXT:
        INC     AX                  ;엔트리 포인터로 AX
        MOV     BX,AX               ;FAT의 실제 offset값을 계산(3/2 * AX)
        SHL     BX,1
        ADD     BX,AX
        SHR     BX,1                ;FAT의 offset값
        MOV     DX,WORD PTR DATA[BX] ;비로소 DX가 FAT값을 취함
        TEST    AX,1                ;엔트리값이 짝수인지 판단
        JZ      EVEN_ENTRY          ;예,최하위 12비트
        MOV     CL,4                ;아니오,최상위 12비트이용
        SHR     DX,CL
EVEN_ENTRY:
        AND     DX,0FFFH            ;최하위 12비트 지정
        CMP     DX,0                ;값이 0인지 판단
        JNE     CHECK_NEXT          ;아니오,다음 클러스터조사
        MOV     DX,AX               ;예,0값을 AX로 이동
        CMP     AX,START_CLUSTER
        JE      POPS            ┐
        MOV     BX,SWITCH_LOC   │
        MOV     CX,[BX]         ├'/A' 와 '/a'옵션 조사
        CMP     CX,SMALL_A      │
        JE      A_OPTION        ┘
        CMP     CX,BIG_A
        JNE     POPS
A_OPTION:
        CALL    PRINT_OUT           ;PROMPT-SECTOR 출력
        JCXZ    CHECK_NEXT
POPS:
        POP     CX
        POP     BX
        POP     AX
        RET
GET_NEXT_ZERO   ENDP

PRINT_OUT       PROC    NEAR
  COMMENT* Returns CX=1 if found the right sector, 0 otherwise *
        PUSH    DX
        ADD     DX,DX               ;섹터번호 계산 (2 *DX)
        ADD     DX,8
        MOV     AL,DISK_DRIVE       ;드라이브지정
        MOV     CX,1
        LEA     BX,PROMPT_SECTOR
        INT     25H                 ;섹터를 읽음
        POPF
        MOV     AH,2                ;출력
        MOV     DL,CR
        INT     21H
        MOV     DL,LF
        INT     21H
        MOV     BX,0
PRINT_LOOP:
        MOV     DL,PROMPT_SECTOR[BX] ;클러스터에서 읽은 문자
        INT     21H                 ;그문자를 출력
        INC     BX                  ;다음 문자로 계속수행
```

```
        CMP     BX,160          ;160문자까지 수행됐는지 판단
        JBE     PRINT_LOOP      ;아니오, 되돌아간다.
        MOV     AH,9            ;예,PROMPT메시지 출력
        LEA     DX,IS_IT_MSG
        INT     21H
        MOV     AH,1            ;키보드로부터 문자지정
        INT     21H
        MOV     CX,1
        CMP     AL,'Y'          ;입력문자 판단
        JE      FOUND
        CMP     AL,'y'
        JE      FOUND
        MOV     CX,0
FOUND:
        POP     DX
        RET
PRINT_OUT       ENDP

PUT_FAT_ENTRY   PROC    NEAR
        COMMENT* Pass FAT cluster number in AX and new entry in DX*
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    DX
        MOV     BX,AX  ;FAT에서 offset번지 계산 (3 * AX/2)
        SHL     BX,1   ;2를 곱한다.
        ADD     BX,AX  ;3 * AX를 얻기위해 AX를 더함
        SHR     BX,1   ;2로 나눈다.그러면 BX는 FAT의 offset값이 됨
        TEST    AX,1                ;짝수인지를 판단
        JZ      P_EVEN_ENTRY        ;예,하위 12비트 지정
        MOV     CL,4                ;아니오,상위 12비트지정
        SHL     DX,CL
P_EVEN_ENTRY:
        OR      WORD PTR DATA[BX],DX   ;FAT에 클러스터번호 기억
        POP     DX
        POP     CX
        POP     BX
        POP     AX
        RET                         ;리턴하고 종료
PUT_FAT_ENTRY   ENDP

CODE_SEG        ENDS
        END     ENTRY
```

〔리스트 2〕 UNDEL.BAS

```
10 '*********************************************
20 '*
30 '*
40 '*   UNDEL.BAS 프로그램
50 '*
60 '*
70 '*------------------------- MAIN PROGRAM -------
80 OPEN "UNDEL.COM" AS # 1  LEN = 1
90 BEEP:PRINT "CREATING UNDEL.COM ..."
100 FIELD #1,1 AS BYTE.$
110 FOR N = 1 TO 532
120 READ BYTE.%
130 IF BYTE.% <> -1 THEN 170
140 LSET BYTE.$ = CHR$(0)
150 FOR I = 1 TO 1536 : PUT #1 : NEXT I
160 GOTO 190
170 LSET BYTE.$ = CHR$(BYTE.%)
180 PUT #1
190 NEXT N
200 CLOSE #1
210 PRINT "UNDEL.COM CREATED."
220 END
230 DATA  233,  112,    6,   40,   67,   41,   49,   57,   56,   56
240 DATA   32,   80,  101,  114,  115,  111,  110,   97,  108,   32
250 DATA   67,  111,  109,  112,  117,  116,  101,  114,   13,   70
```

## 〈용어해설〉

**●의사코드(Pseudocode)**
인간이 알기쉬운 임시의 코드로 자동 프로그래밍의 루틴에 의해서 머신코드로 번역되는 것을 말하며 자동프로그래밍 등을 위하여 쓰인다.

**●인터프리터(Interpreter)**
원시언어의 명령을 하나씩 번역하여 실행하는 프로그램을 말한다. 즉, 계산이 진행됨에 따라 의사코드를 기계코드로 바꿔서 실행하는 것을 말하며 보편적으로 하나의 의사코드에는 기계코드에 의한 하나의 서브루틴이 대응하고 있다.

**●기계어(Machine Language)**
CPU가 그대로 읽어 들여 실행할 수 있는 명령을 말하며 이 명령은 각기 CPU 특유의 부호화된 2진수로 표시되어 있어, 일반적으로 다른 CPU에서의 호환성이 없다. 기계어는 CPU가 처리하기에는 좋지만 사용자 입장에서는 사용법이 까다롭다.

**●인텔리전트 터미널(Intelligent Terminal)**
단말 기능으로서 고성능의 마이크로프로세서 등을 내장한 것을 말한다. 컴퓨터 온라인 데이터 처리 기능과 데이터 통신 기능을 지녀 다기능 단말 장치라고도 불리고 있고 다음과 같은 특징을 가진다.
- 프로그램 처리 기능을 지니며 데이터 처리기능 보유.
- 화일 처리 기능을 지니고 있고 플로피디스크 등의 사용 가능.
- 온라인 단말기로서 통신 기능.
- 워드프로세서 기능을 내장한 기종도 있다.

```
260 DATA 105, 108, 101,  32, 110, 111, 116,  32, 102, 111
270 DATA 117, 110, 100,  32, 100, 101, 108, 101, 116, 101
280 DATA 100,  36,  13,  70, 105, 108, 101,  32,  97, 108
290 DATA 114, 101,  97, 100, 121,  32, 119, 114, 105, 116
300 DATA 116, 101, 110,  32, 111, 118, 101, 114,  36,  13
310 DATA  10,  73, 115,  32, 112,  97, 114, 116,  32, 111
320 DATA 102,  32, 121, 111, 117, 114,  32, 102, 105, 108
330 DATA 101,  32, 104, 101, 114, 101,  63,  32,  40,  89
340 DATA  47,  78,  41,  32,  36,  -1,  46, 160,  92,   0
350 DATA  44,   1, 115,   4, 180,  25, 205,  33, 162,  27
360 DATA   1, 186,   3,   0, 131, 194,   2, 131, 250,  11
370 DATA 114,  11, 180,   9, 141,  22,  28,   1, 205,  33
380 DATA 233, 169,   0, 129,  38,  21,   1,   0,   0, 232
390 DATA 162,   0, 141,  62, 115,   1, 176, 229, 185,   0
400 DATA   4, 242, 174, 227, 215, 190,  94,   0, 187,  11
410 DATA   0,  75, 166, 116, 252, 131, 251,   0, 117, 237
420 DATA  46, 161,  93,   0, 137,  69, 244, 129,  14,  21
430 DATA   1,   1,   0, 232, 118,   0, 139,  69,  14, 163
440 DATA  23,   1, 139,  69,  16, 169, 255,   3, 116,   3
450 DATA   5,   0,   4, 117,  10, 211, 232, 139,  85,  18
460 DATA 117,   6, 211, 226,   3, 194, 163,  25,   1, 186
470 DATA   1,   0, 129,  38,  21,   1,   0,   0, 232,  73
480 DATA   0, 139,  14,  25,   1, 161,  23,   1,  72, 232
490 DATA  99,   0,  59,  22,  23,   1, 116,  18, 180,  19
500 DATA 186,  92,   0, 205,  33, 180,   9, 141,  22,  52
510 DATA   1, 205,  33, 235,  37, 144, 139, 194, 186, 255
520 DATA  15, 131, 249,   1, 116,   3, 232,  62,   0, 232
530 DATA 200,   0, 226, 238, 136,   1,   0, 129,  14,  21
540 DATA   1,   1,   0, 232,   8,   0, 186,   3,   0, 232
550 DATA   2,   0, 205,  32,  80,  83,  81,  82,  87, 160
560 DATA  27,   1, 185,   2,   0, 141,  30, 115,   1, 247
570 DATA   6,  21,   1,   1,   0, 117,   5, 205,  37, 235
580 DATA   3, 144, 205,  38, 157,  95,  90,  89,  91,  88
590 DATA 195,  80,  83,  81,  64, 139, 216, 209, 227,   3
600 DATA 216, 209, 235, 139, 151, 115,   1, 169,   1,   0
610 DATA 116,   4, 177,   4, 211, 234, 129, 226, 255,  15
620 DATA 131, 250,   0, 117, 225, 139, 208,  59,   6,  23
630 DATA   1, 116,  22, 187, 129,   0, 139,  15, 129, 249
640 DATA  47,  97, 116,   6, 129, 249,  47,  65, 117,   5
650 DATA 232,   6,   0, 227, 195,  89,  91,  88, 195,  82
660 DATA   3, 210, 131, 194,   8, 160,  27,   1, 185,   1
670 DATA   0, 141,  30, 115,   5, 205,  37, 157, 180,   2
680 DATA 178,  13, 205,  33, 178,  10, 205,  33, 187,   0
690 DATA   0, 138, 151, 115,   5, 205,  33,  67, 129, 251
700 DATA 160,   0, 118, 243, 180,   9, 141,  22,  79,   1
710 DATA 205,  33, 180,   1, 205,  33, 185,   1,   0,  60
720 DATA  89, 116,   7,  60, 121, 116,   3, 185,   0,   0
730 DATA  90, 195,  80,  83,  81,  82, 139, 216, 209, 227
740 DATA   3, 216, 209, 235, 169,   1,   0, 116,   4, 177
750 DATA   4, 211, 226,   9, 151, 115,   1,  90,  89,  91
760 DATA  88, 195
```

# GW-BASIC의 100%의 활용
# 베이직과 어셈블리 루틴을 결합한다

고석현/항공대 전자과

베이직은 특별히 언급하지 않아도 될 정도로 컴퓨터 사용자라면 누구나 알고 있고 또한 사용하고 있는 대표적인 고수준 언어중의 하나이다. 그런데 베이직은 인터프리터 방식인 이유로 좀 긴 프로그램의 실행시에는 프로그램 처리속도가 현저히 감소된다. 그리하여 이 언어를 기피하고 C나 파스칼 등을 사용하는 유저가 많아졌지만 그래도 베이직은 아직도 많은 일반 유저가 사용하는 가장 보편적이고 배우기 쉬운 언어이다.

그리고 베이직은 외부의 기계어 루틴을 불러서 사용할 수 있는 기능이 있기 때문에 빠른 처리속도가 요구되는 부분을 어셈블리어로 작성해서 실행시키면 처리속도를 향상 시킬 수 있다. 그런데 요즈음은 컴파일러형 베이직도 나와 있어서 베이직에 어셈블리 루틴을 결합할 때 링크(LINK. EXE) 프로그램으로 간단히 링크시킬 수 있지만 인터프리터방식의 베이직에서 어셈블리 루틴을 사용하려면 여러가지의 처리가 요구된다.

대부분의 책들에서 컴파일러형 베이직에서의 어셈블리루틴 결합방법에 대해서는 언급하고 있지만 인터프리터방식은 잘 알려져 있지 않다. 그래서 여기에서는 인터프리터형의 베이직 프로그램과 어셈블리 루틴을 결합하는 것에 대해 간단한 예를 통해서 알아보기로 하겠다.

## 베이직에서 어셈블리 루틴을 호출하는 방법

기계어 루틴을 부르는 베이직의 명령어에는 USR 및 CALL이 있다. 여기에서는 이들 명령어를 이용하여 디스크 드라이브의 리셋 및 드라이브 변경을 할 수 있는 프로그램들을 작성해 보도록 하자.

어떤 유저가 베이직 인터프리터를 기동시켜서 프로그램을 작성하고 디버깅까지 끝나서 프로그램을 디스켓에 저장하려고 막상 세이브(Save) 명령을 내리니 "Disk Full" 및 "Bad Drive" 등의 에러가 발생했다면, 이는 참으로 난감한 일이다. 이 경우에 유저는 당연히 현행 드라이브를 다른 드라이브로 변경하기를 원할 것이다. 그러나 대부분의 베이직 인터프리터가 이 기능을 지원해주지 않기 때문에 프로그램을 포기하고 시스템 모드로 돌아가야만 할지도 모른다.

그러나 지금 드라이브가 A>에 있다면

```
Save "B : EX\BAS.BAS"
```

등의 명령으로 간접적으로 드라이브를 지정해 줄 수도 있지만 이 명령을 실행 후에도 역시 드라이브는 그대로 A>에 있게 된다. 그런데 이처럼 드라이브이름 및 서브 디렉토리를 항상 입력해야 한다는 것은 귀찮고 따분한 일이다. 그러니 베이직으로 옮기고자 하는 드라이브이름을 입력받고, 어셈블리루틴이 이를 실제 행하도록 하는 프로그램을 작성해 보자.

### (1) USR명령에 의한 어셈블리 루틴의 호출

USR명령의 기본 포맷은

```
USR [N] [(argumnent)]
```

이다. n은 0에서 9까지의 숫자중에 하나이며 어느 USR 루틴이 불리워질 것인지를 결정한다. 그리고 Argument는 변수로서 서브루틴에 전달되어질 수나 문자이다. USR명령을 실행하기전에 베이직 프로그램내에서는 반드시 부르고자 하는 기계어 루틴이 수록되어 있는 세그먼트(Segment)와 오프셋(Offset) 주소를 명시해 주어야 한다.

리스트 1의 800번과 900번 문이 바로 세그먼트와 오프셋을 지정하는 명령이다. 그런데 EXE 화일을 만든 다음 디버거(DEBUG. COM) 등으로 확인해보지 않는 한 기계어 루틴이 메모리중의 어느 곳에 로드되는지 알 수 없다. 그러므로 EXE화일이 완성되기까지는 우선 아무런 값이나 넣어두자. 자세한 것은 어셈블리어 작성에서 설명된다.

리스트 1의 프로그램에서 2100번의 USR0(N) 명령이 실행되면 800번과 900번 문에 의해 설정되어진 세그먼트와 오프셋 주소에 의해 결정되어 지는 포인트로 프로그램의 제어권이 넘겨진다. 그런데 베이직 프로그램에서 어셈블리 루틴으로 전달되는 인수 N의 값이 어느 번지에 수록되어 있는가를 알지 못하면 어셈블리 루틴에서 이를 이용할 수 없다.

베이직 언어는 정수, 단정도 및 배정도 실수와 같은 실수연산용 데이터를 FAC라고 하는 장소에 수록하여 관리한다.

〔리스트 1〕 CHDUSR. BAS

```
100 DEFINT A-Z
200 REM ------------------------------------------------------------
300 REM -                                                          -
400 REM -        change drive program use the assembly routine.    -
500 REM -                  designed by go seog hyun.               -
600 REM ------------------------------------------------------------
700 REM set segment and offset value with address in assembly routine.
800 DEF SEG=&H9FE0
900 DEF USR0=0
1000 BLOAD "c:\e\chdusr.bin"
1100 CLS
1200 PRINT
1300 PRINT
1400 PRINT "       a    ----------------------------------------0"
1500 PRINT "       b    ----------------------------------------1"
1600 PRINT "       c    ----------------------------------------2"
1700 PRINT "       d    ----------------------------------------3"
1800 PRINT
1900 PRINT
2000 INPUT "       Enter the number ";N
2100 PP=USR0(N):PRINT
2200 FILES
2300 ON ERR GOTO 3600
2400 LINE INPUT DUMMY$
2500 CLS
2600 PRINT
2700 PRINT
2800 LINE INPUT "   all ok.    more (Y/N) ";MO$
2900 IF MO$="n" THEN GOTO 3000 ELSE 1100
3000 CLS
3100 PRINT
3200 PRINT
3300 PRINT  "good bye ."
3400 CLS
3500 END
3600 REM error subroutine
3700 PRINT
3800 PRINT
3900 PRINT "now it happened the error. check for your command validity."
4000 PRINT
4100 PRINT
4200 LINE INPUT "more (Y/N)";P$
4300 IF P$="y" THEN GOTO 1100
4400 RETURN
```

그렇기 때문에 N도 FAC에 수록되어 지며 어셈블리 루틴이 호출되면 베이직은 어셈블리 루틴이 인수를 사용할 수 있도록 자동적으로 FAC의 세그먼트 및 오프셋 값을 DS와 BX 레지스터에 전송한다. 그러므로 우리가 베이직에서 전달되어진 인수 N을 어셈블리 루틴에서 사용하려면 DS와 BX 레지스터의 값에 의해 정해지는 어드레스에서 값을 얻어내야 한다. 그렇기 때문에 만약에 프로그램 초기의 DS 및 BX 레지스터를 인수전달 목적이 아니고 다른 목적에서 사용하려면 반드시 이를 스택 등에 저장시켜 놓아야 한다.

그런데 BX 레지스터가 가리키고 있는 포인터는 FAC의 시작 어드레스가 아니라 정수가 수록되어 지는 FAC의 5번째 바이트의 어드레스이다. 그러므로 여기에서처럼 정수를 인도받을 때는 단지 BX가 가르키는 어드레스의 값을 얻어내면 되지만 다른 형의 변수를 전달받을 때는 변경이 필요하다.

리스트 1의 프로그램중에서 100번의 Defint A-Z 명령은 A에서 Z까지의 문자를 정수형 변수로 선언하는 것으로 이 프로그램의 실행에 있어서 중요한 명령중의 하나이다.

필자가 처음 이 예제 프로그램을 완성하기 위해서 리스트 1 및 리스트 2의 프로그램을 입력하고 베이직 인터프리터모드에서 실행시켰더니 항상 드라이브가 A>로만 변경되었다. 왜냐하면 그때까지 리스트 1의 100번 문장을 써주지 않았기 때문이었다. 이것을 인식하는데는 오랜 시간이 걸렸으며 이 문제를 해결하기 위해 긴밤(?)을 꼬박 컴퓨터에 매달려 있어야만 했다.

여러분도 이런 점을 주의하기 바라며 "컴퓨터는 현명한 바보"임을 다시한번 실감했다.

다른 언어에서도 어셈블리어와 결합할 때 전달되는 인수의 형은 반드시 확실하게 정의해 주어야만 한다.

베이직 프로그램이 완성 되었으면 이제는 어셈블리어 프로그램을 작성할 차례이다.

어셈블리 루틴이 다른 언어에서 활용되기 위해서는 반드시 Proc far로 작성 되어져야 한다. 리스트 2의 CHDUSR.ASM 프로그램에서 Chd Proc far라는 것은 Chd라는 프로시듀어(Procedure)가 다른 세그먼트에서도 참조될 수 있다고 선언하는 것이다. 그런데 Chd Proc

〔리스트 2〕 CHDUSR. ASM

```
; -------------------------------------------------------------
; -                                                           -
; -                  change drive program .                   -
; -                     designed by go seog hyun              -
; -------------------------------------------------------------


reset_drive       equ 0dh         ;reset drive function number.
change_drive      equ 0eh         ;change drive function number.
dos_function_call equ 21h         ;dos function interrupt number.



; cseg is code segment.

cseg segment para public 'code'
        public chd
        assume cs:cseg

; ------------------------------------------

chd     proc far

start:
        cmp al,2                 ;is it integer ?
        jne return

;reset drive.

        mov ah,reset_drive
        int dos_function_call

;argument 1 set to dx.

        mov ax,[bx]              ;sent integer to ax
        mov dx,ax                ;sent ax to dx

;change drive function
; if dl=0 ,it selected drive a
;    dl=1 , b drive

        mov ah,change_drive
        int dos_function_call

;return process
return:
        ret

chd     endp
cseg    ends
        end start
```

Near로 선언하면 Chd라는 프로시듀어는 이것이 속해 있는 세그먼트 에서만 참조될 수 있다.

그러므로 반드시 Proc far로 선언해야만 한다.

프로그램의 처음 부분을 보면 AL의 값이 2인지를 체크하고 있는데, 이는 베이직으로부터 전달받은 인수가 어떤 형태인지를 판별하는 부분이다.

베이직은 어셈블리 루틴에 인수를 전달할 때 이 인수의 형태에 따라 정해진 값을 AL 레지스터에 돌려 준다. 그러므로 AL 레지스터의 값을 참조하면 어떤 형의 인수가 전달되었는지 알 수 있다. 표 1이 인수 형태에 따른 AL 레지스터의 값이다.

이 어셈블리어 프로그램에서는 정수형 인수가 전달되어야만 하기 때문에 프로그램의 폭주를 예방하기 위하여 AL의 값이 2가 아니면 무조건 베이직 프로그램으로 돌아가도록 작성했다.

여기에서 사용된 MS-DOS의 기능 콜은 Int 21H의 0DH와 0EH이다.

여기에 대한 요약 설명은 표 2에 있다.

드라이브를 변경하기 위해서는 AH에 0EH를 넣고 DL에 드라이브 번호를 넣은 다음 INT 21H를 실행하면 된다.

이처럼 MS-DOS는 필수적인 기능들을 인터럽트를 걸면 실행시킬 수 있도록 지원하고 있기 때문에 어셈블리어 프로그램을 작성하기가 상당히 수월하다. 그런데 필자가 지금 어셈블리어에 대해 이야기 하고자 하는 것은 아니기 때문에 예제 프로그램은 단순한 기능을 가지는 조그만 프로그램을 작성했다. 다른 기능들은 여러분 각자가 재량껏 추가하기 바란다.

이제 어셈블 및 링크를 하고 어셈블리어 프로그램이 메모리의 어디쯤에 수록되는지 DEBUG.COM으로 확인을 해보자. 이 과정은 **리스트 3**에 나와 있다. 그런데 이 어셈블리어 프로그램은 많은 메모리를 차지하는 베이직 인터프리터가 메모리에 로드된 다음에 로드

〔표 1〕 인수형태에 따른 AL레지스터의 값

| AL | 인수형태 |
|---|---|
| 2 | 정수 |
| 3 | 문자열 |
| 4 | 단정도 실수 |
| 8 | 배정도 실수 |

* AL에 2가 들어 있다는 것은 인수가 정수이며, 크기가 2바이트임을 말해주는 것이다.

〔표 2〕 함수 0DH와 0EH의 기능

| 기능번호 | 사용방법 | 전달값 | 기 능 |
|---|---|---|---|
| 0DH | AH←0DH<br>INT 21H | 없음 | 모든 화일 버퍼를 클리어한다. 즉 디스크 드라이브를 리세트시킨다. |
| 0EH | AH←0EH<br>DL←드라이브번호<br>INT 21H | AL←현재 시스템에 장착되어진 드라이브 수 | DL레지스터에 지정된 드라이브가 현행 드라이브로 선택되고 AL레지스터에 접속되어 있는 드라이브 수가 돌려진다. 무효한 드라이브를 선택한 경우, 현행 드라이브는 변경되지 않는다. |

〔리스트 3〕 어셈블 링크 및 기계어 루틴의 수록주소 판정과정

```
C:\R>masm chd;
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

  51618 + 360286 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors


C:\R>link chd/high;

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

LINK : warning L4021: no stack segment


 In basic mode,this is procedure to make .bin

GW-BASIC 2.01
(C) Copyright Microsoft 1983,1984

The TeleVideo Personal Computer Basic  V2.1
Copyright 1984 TeleVideo Systems, Inc.
61530 Bytes free
Ok

def seg=&h9fe0
Ok

bsave "chdusr.bin",0,h11
Ok


C:\E>debug chdusr.exe
-r
AX=FFFF  BX=0000  CX=0011  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=1F17  ES=1F17  SS=9FE0  CS=9FE0  IP=0000   NV UP EI PL NZ NA PO NC
9FE0:0000 3C02          CMP     AL,02
```

되기 때문에 잘못하면 베이직 인터프리터 부분과 겹쳐서 프로그램이 폭주하고 시스템이 정지하게 된다. 그러므로 CHDUSR.EXE 프로그램은 베이직 인터프리터가 로드되는 메모리 부분과 절대로 겹쳐서는 안된다. 따라서 링크시에는 Link/Chdusr/High;의 옵션을 지정해야만 한다.

/High의 옵션은 컴퓨터가 사용할 수 있는 메모리의 되도록 높은 부분에 EXE화일이 로드될 수 있도록 링크시키는 작용을 한다.

CHDUSR.EXE 프로그램이 완성되었으면 리스트 3에서와 같이 Debug의 R 명령으로 메모리의 수록위치 및 그 길이를 알아보아야 한다. 이는 CHDUSR.EXE 화일을 기계어로 저장하는데 꼭 필요한 데이터이다.

프로그램의 수록위치는 컴퓨터 기종과 메모리 용량에 따라 다르기 때문에 리스트 1의 800번에 있는 Def Seg=&H9FE0에서 9FE0의 값은 서로 다를 수 있다.

이 값은 리스트 3에서 디버거로 CHDUSR.EXE 화일을 로드하고 R 명령을 내렸을 때 출력되는 CS의 값으로 설정되어야만 한다. 리스트 3에서 CS는 세그먼트, IP는 오프셋, 그리고 CX는 로드된 프로그램의 크기를 각각 나타낸다.

이상과 같이 디버거로 CHDUSR.EXE를 기동시키면 메모리의 상위부분(여기서는 세그먼트 &H9FE0)에 CHDUSR.EXE가 로드된다.

그런데 베이직에서 기계어를 불러쓰

기 위해서 항상 이와 같은 로드 작업을 해야한다는 것은 귀찮은 일이다. 그러므로 베이직의 인터프리터 모드에서 BSAVE의 명령으로 .BIN형의 화일로 기록해두면 항상 디버거로 로드할 필요가 없이 BLOAD 명령으로 간단히 로드된다.

기계어 루틴을 BIN 형식의 화일로 만들기 위해서는 베이직 인터프리터 모드에서 다음과 같이 명령을 내리면 된다.

```
Def Seg = &H9FE0 ⏎
Bsave " CHDUSR.BIN " , 0 ,&H11 ⏎
```

그러면 CS가 &H9FE0이고 화일 길이가 &H11인 CHDUSR.BIN이라는 2진 화일이 생성된다. 그러므로 리스트 1에서 BLOAD "CHDUSR.BIN"이 실행되면 CS가 &H9FE0이고 화일 길이가 &H11인 기계어가 해당주소에 수록되게 된다.

그럼, 이로써 모든 것이 완성 되었다. 이처럼 인터프리터형의 언어에서 기계어 루틴을 불러쓰는 데는 많은 과정이 필요하고 복잡하지만 그 효과는 대단히 좋다.

이 과정까지 모두 정상적으로 실행됐다면 다음부터는 CHDUSR.BAS를 실행시키면 이 프로그램이 기계어 루틴을 BLOAD 명령에 의해 자동적으로 로드시키게 된다.

리스트 1의 프로그램을 실행시키고 옮기고자 하는 해당 드라이브의 수치를 입력시켜 주면 실제로 드라이브가 변경되고 그 드라이브의 디렉토리가 표시될 것이다.

어떤가! 좀 더 확장시키고 싶지 않는가! 이는 각자에게 맡겨 두겠다.

## (2) CALL 명령을 이용한 어셈블리 루틴의 호출

CALL 명령을 사용할 때와 USR 명령을 사용한 경우의 프로그램은 서로 대동소이하지만 양자간에는 몇가지 차이점이 있다.

우선 USR 명령은 인수를 기계어 루틴에 한번에 하나씩밖에 전달하지 못하지만 CALL문을 사용하면 많은 인수를 한꺼번에 전달하고 또 전달받을 수 있다. 이때 전달하는 인수는 USR명령의 경우처럼 FAC에 저장되는 것이 아니라 인수의 수록주소가 스택에 세트되어 전달되어 진다. 따라서 CALL 명령을 사용하였을 경우에 인수를 참조하려면 스택에서 정보를 얻어내야 한다.

스택에 저장되는 첫번째 인수의 스택 포인터는 (sp+6)이다. **리스트 4**의 CHD.ASM을 보면 "Argument－1 egu bp+6"이라고 씌여있는 것이 있는데 이 부분이 바로 첫번째 인수의 어드레스가 저장되어 있는 스택상의 위치를 나타내주고 있다.

프로그램의 초기에 현재의 bp 레지스터를 스택에 세트시키고 bp를 sp로 변경하였기 때문에 bp+6이나 sp+6은 동일한 포인터를 지시하고 있는 것이다. 인수 전달시 스택의 저장상태는 **표 3**과 같다. 따라서 두번째 인수의 어드레스는 bp+8, 세번째 인수의 어드레스는 bp+10의 위치에 저장되어 있는 것이다. 그리고 나중에 인수를 베이직 프로그램에 되돌릴 때에는 인수의 수록주소에 직접 설정하면 된다. 그러면 베이직 프로그램상에서 간단히 그 인수를 이용할 수 있다.

이 프로그램에서는 인수에 값을 되돌리지는 않지만 만약에 N의 현재값을 되돌리려면 어셈블리어 상에서 (bp+6)에 값을 세트시켜 주면 된다.

CALL 명령을 이용시에는 반드시 프로그램의 초기에 bp 레지스터를 스택에 대피시켜야 한다. 이는 다음에 베이직으로 귀환할 때 bp 레지스터가 필히 기계어 루틴이 호출되기 이전의 상태로 되어 있어야만 하기 때문이다.

```
push bp
move bp , sp
주 프로그램
pop bp
ret x
```

위의 형식은 CALL 명령사용시의 가장 기본적인 형태이다. 그러므로 위의 형식에 의하면 무리없이 동작할 것이다. ret x에서 x는 인수가 몇개이냐에 의해 결정된다. x=(인수의 갯수×2)이다. 위 프로그램은 N이라는 인수 하나만이 전달되고 있기 때문에 Ret 2라고만 하면 만사 OK이다.

**리스트 5**의 베이직 프로그램을 보면 USR 명령을 쓸 때와 몇가지 다른 점을 찾아볼 수 있을 것이다.

세그먼트의 지정은 동일한 방법에 의해서 하지만 오프셋값은 단순히 Count=0으로 넣어서 지정하고 있다.

이는 프로그램이 세그먼트 &H9FE0의 0번지부터 저장되어 있음을 알려주는 것이다. 그리고 USR명령은 PP=USR0(N)처럼 가인수(Dummy Argument)를 써서 루틴을 호출하고 있고 CALL 명령 역시 Count(N)이라는 가인수를 쓰고 있다. 이때 PP나 Count는 단지 주소를 나타내 주는 데에만 이용된다. 그러나 프로그램에 여러개의 far 프로시듀어가 있을 때는 초기에 이들을 Public 지정을 해주고 이들 함수명을 CALL문에 써줘야 한다.

CALL 명령 사용시에도 Bin형식의 화일을 만드는 방법은 USR의 경우와 동일하다. 여기에서도 프로그램의 초기에

**〔표 3〕 CALL 명령 실행시의 스택상태**

| | |
|---|---|
| SP 레지스터의 값 | ←SP |
| 베이직으로 복귀시의 오프셋 주소 | ←SP+2 |
| 베이직으로 복귀시의 세그먼트 주소 | ←SP+4 |
| 첫번째 인수의 수록 주소 | ←SP+6 |
| 두번째 인수의 수록 주소 | ←SP+8 |
| 세번째 인수의 수록 주소 | ←SP+10 |

〔리스트 4〕 CHD. ASM

```
; --------------------------------------------------------------
; -                                                            -
; -                 change drive program .                     -
; -                     designed by go seog hyun               -
; --------------------------------------------------------------


reset_drive       equ 0dh           ;reset drive function number.
change_drive      equ 0eh           ;change drive function number.
dos_function_call equ 21h           ;dos function interrupt number.
argument_1        equ bp+6          ;first argument address from the basic.


; cseg is code segment.

cseg segment para public 'code'
        public chd
        assume cs:cseg

; --------------------------------------------

chd     proc far

start:  push bp             ;save current bp value
        mov bp,sp           ;set bp as sp,so we called value in stack
                            ;       sended basic program.

;reset drive.

        mov ah,reset_drive
        int dos_function_call

;argument 1 set to dx.

        mov si,[argument_1]
        mov dx,[si]

;change drive function
; if dl=0 ,it selected drive a
;    dl=1 ,it selected drive b

        mov ah,change_drive
        int dos_function_call

;return process

        pop bp
        ret 2

chd     endp
cseg    ends
        end start
```

Defint A—Z을 하는 것을 잊지마라. 그렇지 않으면 분명히 오동작을 일으킬 것이다.

## Shell명령의 효과적인 이용

베이직 인터프리터에는 우리가 잘 모르는 강력한 기능들이 많이 있다. 그런데 대부분의 유저들은 이를 잘 몰라서 사용하지 않고 있다. 이중 하나는 Shell 명령인데, 이를 사용하면 베이직 인터프리터를 램에 상주시킨채로 도스 상태로 빠져 나와서 도스 명령들을 실행시키고 다시 돌아갈 수가 있다. 기본원리는 MS-DOS가 제공하는 INT 27H번을 이용하는 것으로 이것을 이용하면 베이직 인터프리터를 효과적으로 이용할 수 있다.

이는 종래의 8비트 컴퓨터에서 돌아가는 베이직을 사용하던 유저들에게는 놀라운 기능일 것이다. 8비트 컴퓨터의 베이직 인터프리터는 도스 명령을 실행시킬 수 없었으며 이는 반드시 도스 상태로 빠져 나가야만 할 수 있었기 때문이다.

그러므로 앞의 프로그램들은 Shell 명령이 없는 베이직 언어에서 휠씬 효율적일 것이다. 그러나 Shell 명령을 사용하려면 속도가 저하되고 디스크 드라이브의 사용시간이 길며, GW-BASIC 프로그램이 수록되어 있는 디스켓에 반드시 시스템 프로그램과 COMMAND.COM이 기록되어 있어야 한다는 제약 조건이 따른다. 그러므로 좀 더 속도를 높이고 기능을 향상시키기를 원하는 사람은 앞의 CHD.ASM을 강화시켜 보기 바란다.

Shell 명령의 형식은 다음과 같다.

```
SHELL [(Command String)]
```

위에서 Command String은 실행시키고자 하는 명령문이다. 이때 명령문으로서 COM.EXE 그리고 BAT 등의 확장명을 가진 프로그램 뿐만 아니라 TYPE, REN 등의 MS-DOS의 내부 명

〔리스트 5〕 CHD. BAS

```
100 DEFINT A-Z
200 REM ----------------------------------------------------------
300 REM -                                                        -
400 REM -         change drive program use the assembly routine.  -
500 REM -                   designed by go seog hyun.            -
600 REM ----------------------------------------------------------
700 REM set segment and offset value with address in assembly routine.
800 DEF SEG=&H9FE0
900 COUNT=0
1000 BLOAD "c:\e\die.bin"
1100 CLS
1200 PRINT
1300 PRINT
1400 PRINT "      a    ----------------------------------------0"
1500 PRINT "      b    ----------------------------------------1"
1600 PRINT "      c    ----------------------------------------2"
1700 PRINT "      d    ----------------------------------------3"
1800 PRINT
1900 PRINT
2000 INPUT "      Enter the number ";N
2100 CALL COUNT(N):PRINT
2200 FILES
2300 ON ERR GOTO 3600
2400 LINE INPUT DUMMY$
2500 CLS
2600 PRINT
2700 PRINT
2800 LINE INPUT "   all ok.    more (Y/N) ";MO$
2900 IF MO$="n" THEN GOTO 3000 ELSE 1100
3000 CLS
3100 PRINT
3200 PRINT
3300 PRINT  "good bye ."
3400 CLS
3500 END
3600 REM error subroutine
3700 PRINT
3800 PRINT
3900 PRINT "now it happened the error. check for your command validity."
4000 PRINT
4100 PRINT
4200 LINE INPUT "more (Y/N)";P$
4300 IF P$="y" THEN GOTO 1100
4400 RETURN
```

령어(Internal Command)도 가능하다.

위의 명령어 라인에 아무 것도 포함되지 않고 단순히 Shell만 입력하면 화면에 시스템 프롬프트가 나타나게 된다. 즉 도스 상태로 가게 된다. 이 시점에서 GW-BASIC으로 돌아가려면 단순히 EXIT만 입력하면 된다. EXIT 명령은 명령어 상에 직접 입력할 수도 있고 배치 화일에 포함되어 있어도 된다. Shell 명령 다음에 단순히 화일명만 입력하면 컴퓨터는 해당 화일명에 대한 .COM, .EXE, .BAT 등의 순서로 그 화일을 검색하여 실행한다. 만약에 명령어를 잘못 입력시키면 COMMAND.COM은 "Bad Command or Wrong File Name"이라는 에러 메시지를 내보낸다.

Shell 명령 이용시에 어떤 것들은 예기치 않은 결과를 일으킬 가능성이 있기 때문에 주의해야 한다. 그러므로 Shell 명령시에 변경시켰던 화일이나 GW-BASIC상에서 오픈 시켰던 모든 화일들은 반드시 클로즈(Close)시켜야 한다. 그리고 다음 디바이스들은 Shell 명령에 의해 변경되어질 수 없다.

인터럽트 콘트롤러(Interrupt Controller), 카운터 타이머(Counter Timers), DMA 콘트롤러 또는 비동기 통신 콘트롤러(Asynchnorous Communication Controllers) 등이다.

위의 사항들을 명심하면 문제없이 Shell 명령을 이용할 수 있을 것이다. **리스트 6**의 COMM.BAS 프로그램은 Shell 명령을 이용하여 GW-BASIC 모드상에서 MS-DOS의 모든 명령을 처리할 수 있도록 작성한 프로그램이고 TIME.BAS 및 DIR.BAS는 각각 시간 및 날짜, 그리고 디렉토리 관리를 특별히 지원해 줄 수 있도록 작성해본 프로그램이다. 만약에 이들 프로그램들을 RENAME(문번호 재지정) 명령을 써서 현재 사용하고 있는 베이직 프로그램의 가장 높은 문번호보다도 높게 초기 문번호를 지정한다면, 체인 머지(Chain Merge) 기능에 의해 이들을 합성시킬 수 있으며, 이렇게 되면 수시로 이들을 호출하여 MS-DOS의 명령을 베이직 모드상에서 직접 실행시키면서 현재의 프로그램 또한 실행시켜 볼 수 있을 것이다. 이상에서 본인이 사용하였던 소프트웨어는 GW-BASIC, MASM.EXE, LINK.EXE, DEBUG.COM이고, 사용 OS는 MS-

〔리스트 6〕 COMM. BAS

```
100 REM ---------------------------------------------------------------
200 REM -       system support program  through shell command.        -
300 REM -                                                             -
400 REM -              designed by go seog hyun                       -
500 REM ---------------------------------------------------------------
600 CLS
700 LINE INPUT "command : ";C$
800 IF C$="n" THEN GOSUB 3000
900 CLS
1000 SHELL C$
1100 ON ERR GOTO 3700
1200 LINE INPUT DUMMY$
1300 CLS
1400 PRINT
1500 PRINT "if you press the ctrl-c,you will return to the gw-basic mode."
1600 PRINT "and ifyou enter the 'n',it terminates normally."
1700 PRINT "if you don't press any command,you will go to the system mode."
1800 PRINT "if you want to return to the gw-basic mode from system mode"
1900 PRINT "you have to enter the 'exit'"
2000 PRINT
2100 LINE INPUT "press any key to take more command. ";DUMMY$
2200 IF DUMMY$="n" THEN GOSUB 3000
2300 CLS
2400 GOTO 700
2500 END
2600 REM ---------------------------------------------------------------
2700 REM -                                                             -
2800 REM -              final message output subroutine.               -
2900 REM ---------------------------------------------------------------
3000 CLS
3100 PRINT
3200 PRINT
3300 PRINT "system support program end .     good luck to your study"
3400 PRINT
3500 PRINT
3600 END
3700 REM error subroutine
3800 PRINT
3900 PRINT
4000 PRINT "now it happened the error. check for your command validity."
4100 PRINT
4200 PRINT
4300 LINE INPUT "more (Y/N)";P$
4400 IF P$="y" THEN GOTO 600
4500 RETURN
```

DOS Ver3.3이며 컴퓨터는 IBM-PC Turbo XT이다.

지금까지의 내용이 여러분들이 베이직을 좀 더 효율적으로 사용하는데 도움이 되기를 바란다. 또 참고할 것이 있는 분은 GW-BASIC Reference Manual을 참고하기 바란다.

〔리스트 7〕 TIME·BAS

```
100 REM -----------------------------------------------
200 REM -                                             -
300 REM -     setting time and date in basic mode     -
400 REM -                                             -
500 REM -           designed by go seog hyun          -
600 REM -----------------------------------------------
700 CLS
800 PRINT
900 PRINT "     ---------------------------------------"
1000 PRINT
1100 PRINT "    time        ----------------------------1"
1200 PRINT "    date        ----------------------------2"
1300 PRINT "    quit        ----------------------------3"
1400 PRINT
1500 PRINT "     ---------------------------------------"
1600 PRINT
1700 PRINT
1800 INPUT "command";CM
1900 ON CM GOSUB 2200,3200,5000
2000 ON ERR GOTO 4200
2100 GOTO 400
2200 CLS
2300 REM time setting
2400 PRINT
2500 PRINT
2600 LINE INPUT "     time(00:00:00): ";TI$
2700 IF TI$="n" THEN RETURN
2800 TIME$=TI$
2900 ON ERR GOTO 4200
3000 CLS
3100 RETURN
3200 CLS
3300 REM date setting
3400 PRINT
3500 PRINT
3600 LINE INPUT "       date(mm/dd/yyyy): ";DA$
3700 IF DA$="n" THEN RETURN
3800 DATE$=DA$
3900 ON ERR GOTO 4200
4000 CLS
4100 RETURN
4200 REM error subroutine
4300 PRINT
4400 PRINT
4500 PRINT "now in happened the error.
           check for your command validity."
4600 PRINT
4700 LINE INPUT "more(y/n)";P$
4800 IF P$="y" THEN GOTO 700
4900 RETURN
5000 REM -----------------------------------------------
5100 REM -                                             -
5200 REM -      final message output subroutine.       -
5300 REM -----------------------------------------------
5400 CLS
5500 PRINT
5600 PRINT
5700 PRINT "  system support program end .
              good luck to your study"
5800 PRINT
5900 PRINT
6000 END
```

〔리스트 8〕 DIR. BAS

```
100 REM -----------------------------------------------
200 REM -                                             -
300 REM -     directory management subroutine         -
400 REM -        designed by go seog hyun             -
500 REM -----------------------------------------------
600 CLS
700 PRINT "  make directory   -------------------- 1"
800 PRINT "  remove directory -------------------- 2"
900 PRINT "  change directory -------------------- 3"
1000 PRINT "  quit             --------------------4"
1100 PRINT
1200 INPUT "command: ";A
1300 ON A GOSUB 1800,2500,3200 ,4900
1400 ON ERROR GOTO 4100
1500 CLS
1600 GOTO 700
1700 REM error subroutine
1800 CLS
1900 REM      make directory
2000 LINE INPUT "enter the path & directory
                 name( mkdir ): ";MK$
2100 IF MK$="n" THEN RETURN
2200 MKDIR MK$
2300 CLS
2400 RETURN
2500 CLS
2600 REM     remove directory
2700 LINE INPUT "enter the path & directory
                 name (rmdir):";RM$
2800 IF RM$="n" THEN RETURN
2900 RMDIR RM$
3000 CLS
```

```
3100 RETURN
3200 CLS
3300 REM     change directory
3400 LINE INPUT "enter the path & directory name
               ( change directory ): ";CH$
3500 IF CH$="n" THEN RETURN
3600 CHDIR CH$
3700 SHELL "dir/w"
3800 LINE INPUT DUMMY$
3900 CLS
4000 RETURN
4100 REM error subroutine
4200 PRINT
4300 PRINT
4400 PRINT "now it happened the error.
        check for your command validity."
4500 PRINT
4600 LINE INPUT "more(y/n)";P$
4700 IF P$="y" THEN GOTO 600
4800 RETURN
4900 REM ---------------------------------------------
5000 REM -                                           -
5100 REM -       final message output subroutine.    -
5200 REM ---------------------------------------------
5300 CLS
5400 PRINT
5500 PRINT
5600 PRINT "system support program end .
        good luck to your study"
5700 PRINT
5800 PRINT
5900 END
```

# VDISK의 제작

컴퓨터의 기억장치는 주기억장치와 보조기억장치로 나눌 수 있다. 주기억장치는 중앙처리장치와 직접 데이터를 교환하기 때문에 속도가 빠르다. 그러나 보조기억장치는 디스크 드라이브라는 장치를 이용하여 주기억장치에 전송하여 중앙처리장치와 교환하기 때문에 상당한 시간이 지연된다.

그래서 MS-DOS에서는 VDISK(Virtual DISK)라는 방법을 만들어 이러한 문제를 해결하였다.

## VDISK

### 1) 개 념

VDISK의 개념은 MS-DOS V.2.×× 이후의 버전에 도입되었다. VDISK의 장점은 주기억장치의 일부를 떼어서 마치 내부에 드라이브를 하나 더 달은 것처럼 사용할 수 있다는 점이다.

이렇게 함으로써 보조기억장치를 주기억장치의 속도만큼 빠르게 사용할 수 있다. 또, 디스크를 계속 사용하게 되면 접촉부분의 마모가 생기고 회전장치에 문제가 생기곤 한다. 이럴 때 VDISK를 사용하면 이러한 점들을 예방할 수 있다.

또, 여러 개의 VDISK를 만들어서 서로 다른 화일들을 나누어서 사용할 수도 있다.

그러나 VDISK를 사용하면 주기억장치의 용량을 떼어서 사용하기 때문에 주기억장치의 크기가 감소하여 줄어든 주기억장치의 크기보다 큰 데이터는 주기억장치에 적재할 수 없다.

또 〈RESET〉 키나 〈alt〉+〈ctrl〉+〈del〉 키를 누르거나 전원이 꺼지면 모든 주기억장치의 내용이 없어지기 때문에 V-DISK에서 사용된 화일을 모두 보조기억장치에 저장시켜야 한다는 단점을 가지고 있다.

### 2) 설 치

MS-DOS시스템 디스켓을 보면 VDISK.SYS라는 화일이 있다. VDISK의 제작은 이 화일을 이용해서 만드는데 먼저 CONFIG.SYS라는 화일을 만들어야 한다.

CONFIG.SYS는 시스템을 구성하는 명령들의 목록을 포함하고 있는 화일이다. 전원을 올렸을 때, 컴퓨터는 A드라이브의 목록들을 조사하여 CONFIG.SYS를 찾는다. 만약, CONFIG.SYS가 존재하지 않으면 시스템에서 정의된 장치를 설치한다. 그러나 CONFIG.SYS가 존재하면 CONFIG.SYS에 있는 명령에 따라 장치를 설치한다.

CONFIG.SYS에서 VDISK를 만드는 명령형식은 다음과 같다.

```
device = [d:]vdisk.sys [comment][bbb][comment][sss]
                    [comment][ddd][/E[:m]]
```

| | |
|---|---|
| [ ] | : 생략해도 좋음 |
| d | : VDISK.SYS가 있는 드라이브를 지정 |
| comment | : 'buffer size=', 'sector size=', 'directory entries=' |
| bbb | : 버퍼크기 |
| sss | : 섹터크기 |
| ddd | : VDISK가 설치된 디렉토리내의 화일의 최대갯수 |
| E | : VDISK를 확장할 때 사용(IBM PC/AT에서 사용) |

버퍼크기는 VDISK가 주기억장치(640KB)를 떼어쓰는 크기와 같다. 버퍼크기가 1K보다 작으면 자동으로 64KB만큼 VDISK가 설치된다. 버퍼크기가 생략되면 자동적으로 64KB만큼 VDISK가 설치된다.

섹터 허용크기는 128,256,512바이트이다. 만일 사용자의 VDISK가 비교적 작은 화일들을 갖게 하려면 사용하지 않는 공간을 최소화하기 위하여 보다 작은 섹터를 쓰는 게 좋다. 보다 큰 섹터크기를 쓰면 쓸수록 성능이 개선된다. 만약 섹터크기가 생략되면 자동적으로 128바이트가 된다.

(예) device = vidsk.sys 240 128 64

VDISK의 크기, 즉 주기억장치에서 떼어내는 크기는 240KB이다. 섹터의 크기는 128B이다. VDISK에서 사용할 수 있는 총 화일의 갯수는 64개이다.

config.sys를 만드는 방법은 다음과 같다.

```
A> COPY CON CONFIG.SYS
DEVICE = VDISK.SYS BUFFER SIZE = bbb SECTOR SIZE = sss
        DIRECTORY entries = ddd
<ctrl>+Z
A>
```

여기서 컴퓨터를 처음부터 다시 작동시키면,

```
VDISK Version x.xx virtual disk C:
      Buffer size:      bbb KB
      Sector size:      sss
      Directory entries: ddd
```

라는 메시지가 나오고 DOS가 실행된다. 이제 VDISK가 만들어졌다. 메시지에는 디스크 c : 라고 나타내고 있다. PC에 달려 있는 A와 B드라이브 다음으로 C 드라이브에 VDISK를 할당하게 된다. 이제부터 주기억장치의 일부를 떼어서 VDISK를 드라이브처럼 사용할 수 있다.

여러 개의 VDISK를 만들고 싶으면 CONFIG. SYS에서 디바이스문을 여러 개 쓰면 디바이스문의 갯수 만큼 C, D, E.. 순으로 할당된다.

VDISK를 사용하려면 필요한 화일을 VDISK로 할당된 드라이브에 복사시켜야 한다. 만약 VDISK가 C 드라이브에 할당되었으면, C드라이브로 바꾸어 주어야 한다.

A>C : ↵ 하면 C>로 바뀐다. 마치 드라이브를 하나 더 설치한 것처럼… VDISK는 디스크 드라이브를 단지 주기억장치에 저장시킨다는 것만 다르고 사용방법은 똑 같다.

VDISK와 보통 디스크 드라이브와 비교해 보면, 디스크 드라이브에 있는 데이터를 실행시킬려면 우선 디스크 드라이브의 내용을 읽어서 주기억장치로 저장한 후 저장된 데이터를 실행시키는 반면에 VDISK에서는 주기억장치를 떼어서 쓰기 때문에 바로 실행시킬 수 있어서 디스크 드라이브와는 엄청난 속도의 차이가 난다.

앞에서 설명했듯이 VDISK에 있는 화일들은 작업이 모두 끝났을 때, 전부 보조기억장치로 복사시켜 놓아야 다음에 다시 사용할 수 있다. 이러한 단점에도 불구하고 빠른 속도와 드라이브 사용을 줄임으로써 디스크 드라이브의 수명을 연장시킬 수 있다. 그러므로, VDISK에는 자주 쓰이는 프로그램을 저장하는 것이 좋다. PC

# A/D 콘버터

최승욱/태성전산

처음 우리나라에 마이컴이 상륙했을 때 컴퓨터에 대한 기대는 대단했었다. 많은 사람들이 컴퓨터만 있으면 컴퓨터가 밥도 자동으로 하고 빨래도 해주는 등 우리 일상생활의 모든 것을 자동으로 처리해 주는 만능기계로 생각했다. 그런데 막상 컴퓨터를 사고보니 밥은 커녕 가계부 정리도 못하는 것에 크게 실망하여 타오르던 컴퓨터 열기는 사그라들고 컴퓨터의 재인식이라는 현재의 단계에 까지 와있다. 그러나 실제로 컴퓨터는 밥도 빨래도 모두할 수 있으며 가계부 정리도 할 수 있다.

물론 컴퓨터는 손과 발이 없기 때문에 컴퓨터가 직접 돌아다니며 가정부처럼 집안일을 돌볼 수는 없다. 하지만 전기적인 신호로서 전기밥솥이나 전기세탁기를 제어할 수 있다. 이와 같이 컴퓨터가 자신의 세계와 외부의 세계를 연결시켜 주는 것을 인터페이스(Interface)라고 하며, 본고를 통하여 인터페이스를 실제적으로 실현시켜보고자 한다.

과거에 컴퓨터에 대한 큰 기대로 컴퓨터를 구입했다가 실망하여 먼지만 쌓인 자신의 시스템을 이 기회에 과거의 기대를 실현시킬 수 있는 시스템으로 바꾸어 보기 바란다.

## 정보의 수집

퍼스널컴퓨터를 사용하여 외부의 세계를 제어하기 위해서는 당연히 외부의 세계에 대한 정보를 알아야 한다.

인간의 경우 어두워지면 눈을 통하여 어두움을 감지하여 전등을 켠다. 이와 마찬가지로 PC를 통해 외부의 세계를 제어하기 위해서는 외부에서 동기적, 비동지적으로 발생하는 외부의 정보를 컴퓨터가 읽어 들여서 그것에 대응하는 행동을 취하게 하여 컴퓨터가 외부세계를 어떠한 특정상태에 도달하게끔 제어를 하게 해야 한다. 그런데 컴퓨터 그 자체는 계산기이지 계측기가 아니기 때문에 직접 외부의 정보가 되는 모든 물리량을 판독하는 것은 어려운 일이다. 이런 이유로 컴퓨터와 외부세계 사이를 연결할때 두가지 필요한 과정이 있다.

첫째, 외부의 물리량을 전기적인 신호로 바꾸어 주는 것이 필요하고, 둘째, 컴퓨터로 그 바뀌어진 전기신호를 읽어 들여야 한다. 물론 외부의 정보를 임의의 수치로 환산하여 키보드를 통하여 컴퓨터에게 전달할 수도 있지만 그것은 자동화란 측면에 위배된다.

일반적으로 컴퓨터는 외부의 물리량인 연속적인 아날로그신호를 직접 다루지 못하고 모든 것을 전기적신호와 같은 디지탈신호로 변환하여 처리하는 것이 보통이다. 그래서 앞에서 설명한 두번째의 과정을 충족시켜 주는 아날로그-디지탈 변환기(이후는 ADC라 한다)를 알아야 한다. 또한 이러한 ADC를 사용하기 위해서는 부수적으로 첨가되는 여러가지 회로를 알아야 하는데 이에 대해서도 간단히 설명하겠다.

## 외부 데이터의 입력

### (1) 비동기적 데이터입력

우리의 주변에서 발생하는 거의 대부분의 데이터는 비동기적이며 연속적인 물리량이다. 그러나 유감스럽게도 인간이 사용하는 언어는 한계성을 가지고 있어서 사실상 아날로그적인 표현은 어렵다. 예를 들면 길이를 말하는 경우에 '이것은 저것의 몇배이다'라는 식으로 말하며 이때 이것은 측정대상이고 저것은 약속된 단위길이가 될 것이다.

비동기적 데이터의 입력은 어떤 의미에서는 컴퓨터와의 관련성을 유보하고 생략하여도 괜찮을 정도로 입력과 컴퓨터가 별도로 동작한다. 그러나 주변입력에 대한 처리의 응답속도를 빠르게 하기 위해서는 부가되는 장치가 더 필요하다.

### (2) 데이터 입력방법

컴퓨터가 외부의 데이터를 읽기 위해서는 여러가지 방법이 있는데 다음과 같다.

1. 트랜스듀서의 신호를 직접 입력
2. 트랜스듀서의 신호를 ADC로 입력
3. 트랜스듀서의 신호를 V-F를 사용하여 입력

#### (1) 트랜스듀서를 이용한 방법

이것은 매우 디지탈적인 방법이며 컴퓨터의 고유속성과 일치한다. 이때 트랜스듀서로 부터의 신호는 "1"이나 "0"인 2진 데이터로써 읽혀지는데 그것에 관하여는 **그림 1**에 나타나 있다. 이 경우 트랜지스터 베이스에 광전자 유무에 따라 출력 F는 "1"이나 "0"인 출력을 내보낸다. 즉 출력 F는 외부의 빛의 유무에 따라 1비트의 2진 데이터

를 발생시킨다. 그림에서의 VR은 임계 전압조절용이다.

(2) 아날로그-디지탈 변환기의 사용

이것은 가장 보편적인 방법이며 어느 정도의 수준에서 컴퓨터가 비교적 외부의 데이터를 정확히 읽어내기 위해서는 이 방법을 반드시 채택해야 하며 본고의 촛점이기도 하다. 따라서 여기에 대해서는 상세히 설명하고자 한다.

**●ADC의 종류**

(1) **계수형 ADC**

① 램프형 ADC
② 이중 적분형 ADC
③ 전하평형형 ADC

(2) **비교형 ADC**

① 축차비교형 ADC
② 추종비교형 ADC
③ 병렬비교형 ADC

**●ADC의 특성**

(1) **램프형 ADC**

**그림 2**의 다이어그램은 램프형 ADC를 간략히 나타낸 것이다. 그림에서 보는 바와 같이 $S_1$과 $S_2$는 서로 보수(Complement)적으로 작동하기 때문에 $S_2$가 ON 되면 콘덴서 C는 완전히 방전된다. 두번째 인버터는 "L"이 될때 C를 방전시킨다. 이후 $S_1$이 ON 되면 입력 기준전압을 적분하기 시작한다. 그 적분파형이 삼각파형인데 이것이 램프파라 하는 ADC에서 사용하는 기준전압이 된다. 램프형 ADC의 이름이 바로 여기서 유래한 것이다.

이때 기준전압은 ADC의 기준이 되어야 하므로 매우 정확해야 하는데 온도변화에 의해 R과 C에 변화가 올 수 있고 이것은 램프형 ADC의 단점이기도 하다. 발생한 기준 삼각파는 입력전압 $V_i$와 비교기 B에서 비교되어 R-S 플립플롭을 작동시킨다.

비교기 A는 카운터의 개시 신호용이다. 적분기의 출력은 약간 마이너스 측에서 발생하는데 플러스 방향으로 상승되는 지점에서 비교기 A의 출력이 "H"가 된다. 이것은 R-S 플립플롭을 세트시켜 클럭 CLK(t)를 입력 인에이블(Input Enable)시켜 카운터를 동작시킨다. 이때 입력전압과 기준전압을 비교하여 $V_i$보다 크게 되면 R은 "H"로 세트되고 이것은 클럭 CLK(t)를 디스에이블시킨다. 이때 $\overline{Q}$에서는 EOC(End of Conversion)을 발생시켜 변환과정을 마친다. **그림 3**은 설명의 이해를 위한 것으로 게이트의 펄스파는 카운트를 하는 클럭을 의미한다.

(2) **이중적분형 ADC**

듀얼 슬로프형이라고도 불리우는 이 ADC는 비교적 널리 쓰이고 있다.

이것은 입력전압의 적분치에 대한 기준전압의 적분치에 대한 비로써 변환을 실현한다. 또한 R과 C에 의한 적분시의 오차에 의한 왜곡이 발생하지 않아서 단일 램프변환형에 대해서 오차가 더 작다. 또한 입력전압은 입력전압의 평균치로써 적분되기 때문에 입력전압의 약간의 변동에도 영향을 받지 않아 노이즈에 대한 내성을 갖는다.

동작의 순서는 카운터를 리세트한 후에 입력클럭을 카운트하는 카운터의 오버플로우가 나올때까지 $V_i$를 적분한다. 이때 카운터의 오버플로우시 발생하는 신호에 의해 기준전압이 마이너스방향으로 적분되고 이것이 제로크로스될 때까지 카운트한다. 이때 제로크로스될 때의 최종적인 카운터값이 입력전압에 비례한 값이 된다. **그림 4**는 이중적분형의 로직 다이어그램이다.

(3) **전하평형형 ADC**

전하평형형 ADC는 모놀리딕 IC에 널리 이용되고 있고 고정확도를 얻을 수 있다. 또 이것은 델타 시그마형의 변조기에도 응용될 수 있는 구조로 되어 있다.

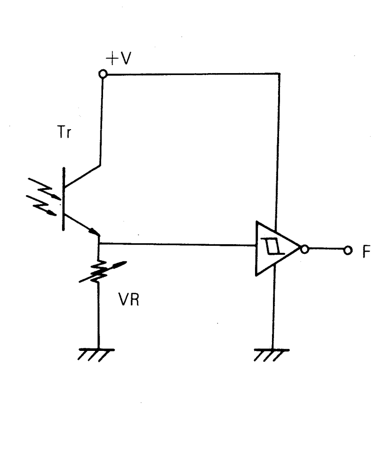
**그림 5**를 보면 입력전압에 비례하는 펄스열을 만들고 이를 카운터의 클럭으로 받아서 그 값을 읽어서 입력전압을 측정하는 것이 나타나 있는데 이것은 적분출력을 가능케하는 이미지너리 어스(Imaginary Earth)에 의한 개념에서 발생된 것이며, C에 충전된 전하가 입력 $V_i$와 단속하고 있는 $I_{ref}$로서 상쇄하도록 조작을 반복한다. 이때 S의 ON, OFF 횟수가 $V_i$에 비례한다.

(4) **축차비교형 ADC**

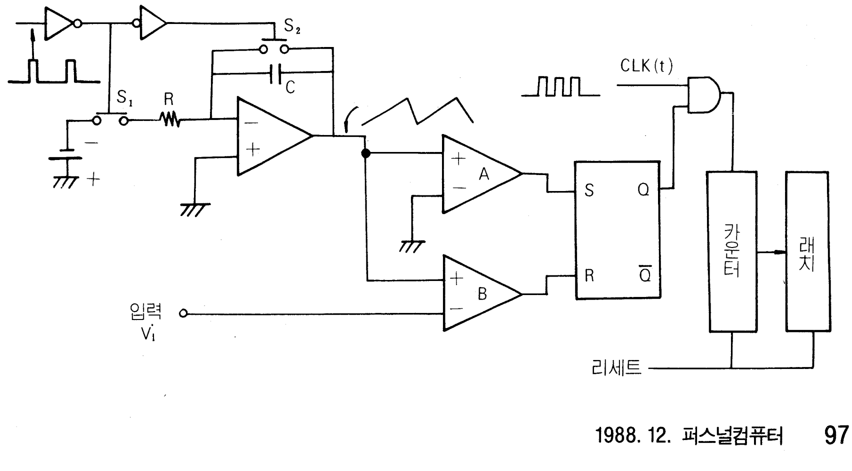
이 구조는 ADC에서 널리 쓰여지는 구조이고 변환속도가 비교적 짧다.

축차비교형 ADC의 변환속도는 축차

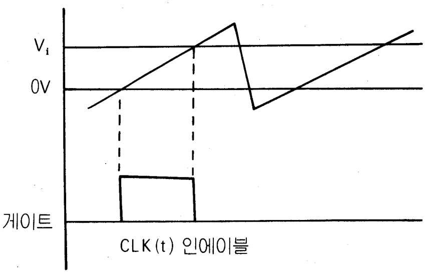
〔그림 1〕 트랜서듀서를 이용한 데이터입력



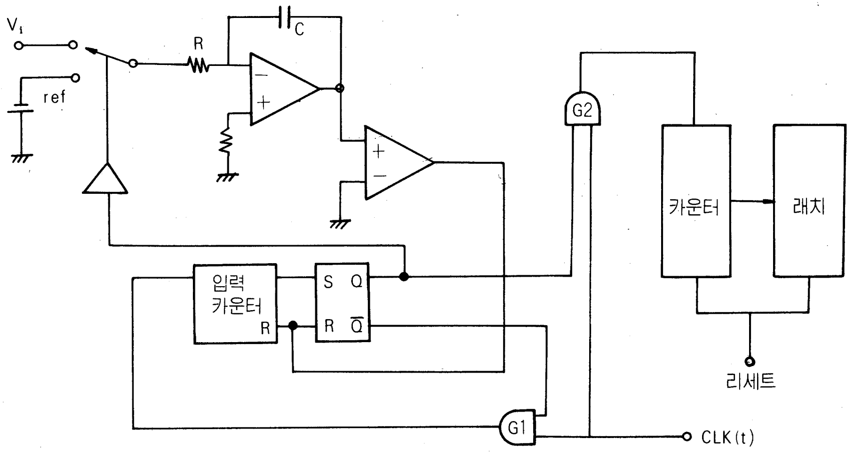

〔그림 2〕 램프형 ADC



〔그림 3〕 변환과정을 펄스파로 표현




〔그림 4〕 이중적분형 ADC의 로직 다이어그램




비교 레지스터의 비트 수만큼 일어난다. 즉 3비트 SAR을 갖는 경우 단위 클럭이 3개 필요하다. 여기서 SAR은 Successive Approximation Registor의 약자이며 이러한 것을 채용했다 하여 SAC라고 하기도 한다. **그림 6**은 이러한 축차비교형 변환기를 보인 것이다. 이것의 동작 이해를 위해 **리스트 1**에 동작을 나타내는 프로그램을 보였다. 루프의 도는 횟수가 n비트 SAR을 의미한다.

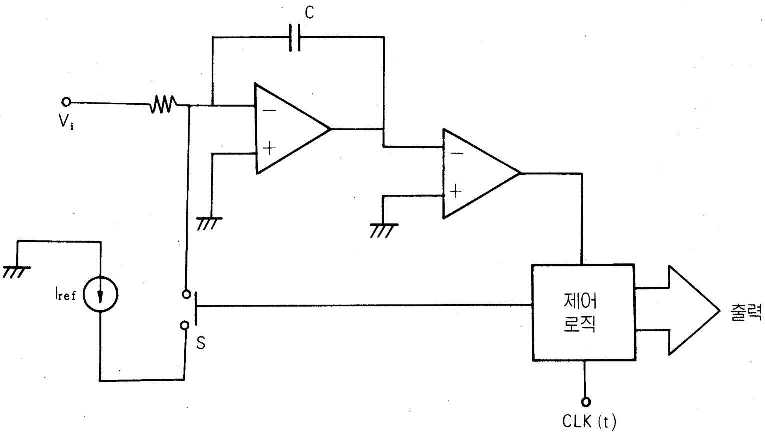
(5) **추종비교형** ADC

추종비교형 ADC는 카운터의 구성이 업/다운 카운터의 구성으로 되어 있다. 즉 업/다운 카운터는 콘버터의 비교기에 의해 업/다운이 제어되는 것으로 입력전압이 출력전압보다 작으면 업 카운트를 하고 반대의 경우는 다운카운트한다. 따라서 입력전압의 변동이 급격할 경우 이를 변환하는 것은 어려우며 완만하게 변동하는 신호의 변화에 적당하다(**그림 7** 참고).

(6) **병렬형** ADC

병렬형 ADC는 고속 ADC의 대표적인 것으로서 다단형 비교기를 병렬로 접속하여 각각의 비교기에 기준무게를 설정하여 들어온 입력을 각 비교기의 기준입력에 대해 한꺼번에 비교하여 디지탈코드로 변환하는 구성으로 되어 있다. 예를 들면 VTR의 경우나 화상처리를 할 경우 순간적으로 프레임을 고속으로 저장해야할 경우가 생기는데 이런 경우 병렬형 ADC를 사용하여 처리

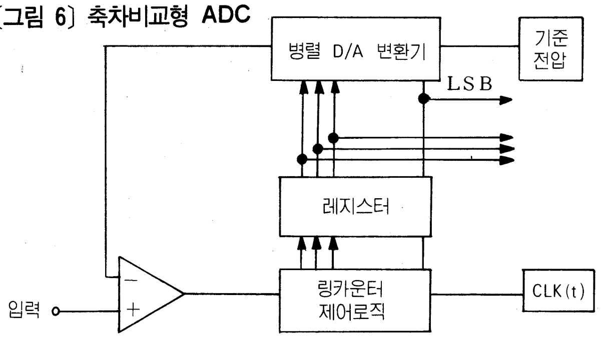
〔그림 5〕 전하평형형 ADC의 로직 다이어그램



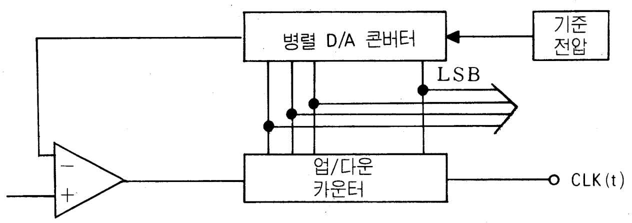

〔리스트 1〕

```
100 ' Programed by C.S.W 88/10/24  2:43 am
110 ' Successive approximation converter  where n bit SAR is built.
120 ' Result is left on SAR.
130 '
140 ' Note that b(n) means SAR's nth digit in binary number.
150 '
160 N=3                              :' n bit SAR
170 INPUT "input voltage ";V         :' input voltage Vi
180 SAR=0                            :' init SAR
190 '                                :'
200 FOR BIT=N-1 TO STEP -1           :' Repeat n=n-1
210 SAR=SAR OR 2^BIT                 :'        Set b(n)
220 IF SAR>V THEN SAR=SAR XOR 2^BIT  :'        if SAR>Vi then Repeat b(n)
230 NEXT BIT                         :' Until n=0
240 '
250 PRINT V,SAR                      :' print out input voltage and result
260 GOTO 170
```

〔그림 6〕 축차비교형 ADC




〔그림 7〕 추종비교형 ADC




할 수 있다(그림 8 참고).

**(3) V-F 변환기의 사용에 의한 ADC**

**그림 9**는 적분기와 원쇼트 멀티 바이브레이터로 구성되어 있는 V-F 콘버터의 원리적 회로이다. 여기서 V는 전압을 의미하며 F는 주파수를 의미하는 것으로 임의 전압의 크기를 그에 상당하는 주파수로 변환하는 것이다. 따라서 출력 주파수를 측정하므로써 입력전압을 알 수 있다. 그림 9의 회로에서 $C_1$에 병렬로 연결된 트랜지스터 $Tr_1$은 베이스에 플러스 전압을 가할때 바이어스 된다. 이때 $C_1$은 단락된다. $E_f$의 출력은 평시에는 $A_2$의 플러스 입력단자에 마이너스의 전압이 가해지고 있기 때문에 마이너스 방향의 $E_f$ 출력이 나온다. 따라서 $Tr_1$은 역바이어스가 걸리고 $A_1$은 적분기로서 동작하게 된다. $E_1$의 전압을 적분하면 A점은 점점 마이너스측으로 내려간다.

이것이 비반전 입력단자의 마이너스 값과 거의 같아질때 $A_2$는 트리거되어 $C_2$, $R_2$~$R_4$의 시정수의 펄스를 출력하고 이것은 $Tr_1$에 순바이어스를 가하여 $C_2$를 리세트 시키고 다시 $A_2$의 출력은 일정 시간후에 마이너스가 되어 적분기를 동작시킨다. 이 동작의 반복에 의해 전압-주파수 변환이 행하여 진다.

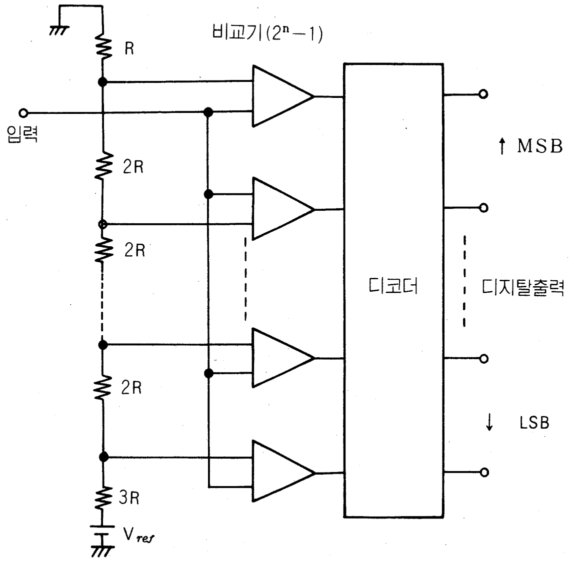
# 부수적 회로

실제적으로 ADC를 이용하여 외부의 데이터를 읽기 위해서 단순히 ADC만으로는 매우 불안정한 데이터를 판독하게 된다. 특히 SAC의 경우 샘플&홀드회로 없이는 완전히 엉터리 데이터를 읽게 되므로 주의가 필요하다. 또한 트랜스듀서의 비직선성에 의해 이를 선형화해 줄 수 있는 리니어라이저의 필요성과 미소신호 ADC의 경우 전치 입력단의 정밀한 설계가 요구된다.

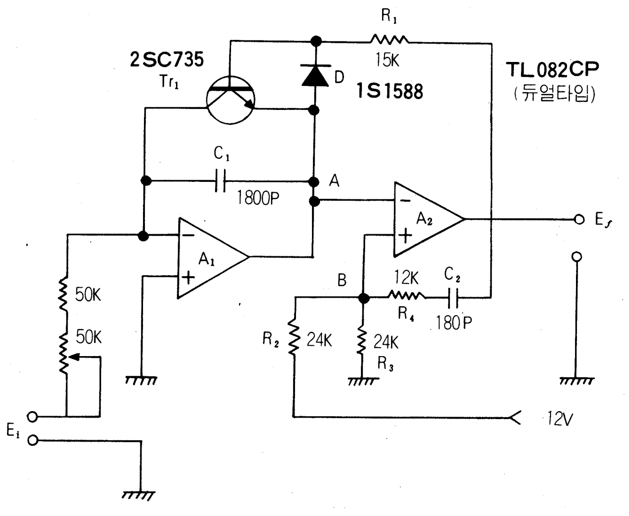
특히 이러한 경우 전치 입력단의 정밀도에 의해 ADC의 정밀도가 결정될 만큼 중요하며 입력신호의 고조파 성분을 충분히 억제할 수 있는 필터가 필요하다.

## 1. 샘플&홀드회로

〔그림 8〕 병렬비교형 ADC



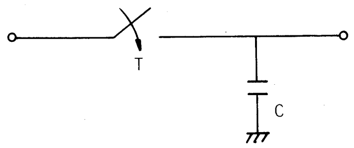

〔그림 9〕 V-F변환기의 사용에 의한 ADC



샘플&홀드 회로는 ADC에 있어서는 약방의 감초와 같은 존재이다. 사실상 비교적 고속을 요구하는 ADC의 경우 이를 첨가하나 이론적으로 안정된 ADC를 위해서는 반드시 부착되어야 한다. 이러한 샘플&홀드 회로는 ADC고유의 변환시간 기간동안 입력이 변하면 않되기 때문에 필요하다.

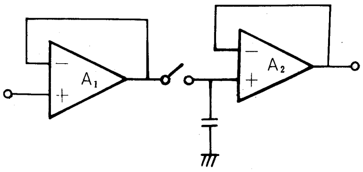
실제로 이상적인 샘플&홀드 회로의 구성은 어려우나 차동증폭기를 이용하여 근사적으로 구현할 수 있다. 일반적으로 차동증폭기의 입력임피던스는 대단히 커서 $10^5$정도 큰값을 가지므로 입력임피던스의 크기가 무한대(∞)로 볼 수 있고 출력임피던스는 0Ω에 가깝다. 이러한 특성을 갖는 차동증폭기를 볼테지 폴로워로 구성하여 홀드회로 즉 캐퍼시터의 양단에 접속하므로써 실현된다. **그림 10**은 샘플&홀드의 원리적 설명이고 **그림 11**은 실제적으로 차동증폭기를 사용하여 구성한 경우이다.

〔그림 10〕 샘플, 홀드의 기본원리



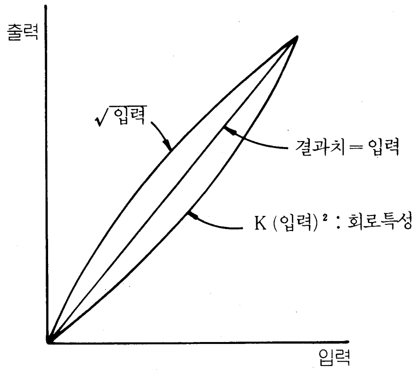

스위치가 닫히는 순간 C가 충전되어 입력값이 저장된다

〔그림 11〕 샘플, 홀드를 차동증폭기를 이용하여 구성

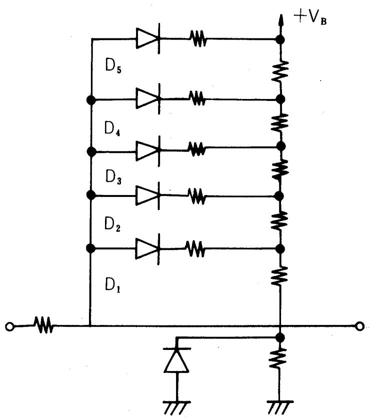



실제로는 C와 $A_2$의 입력 임피던스로 시정수가 결정된다

〔그림 12〕




(a) 입력에 대한 출력값




(b) 절선근사를 위한 리니어라이저
(다이오드 사용)

## 2. 리니어라이저

이것은 비선형적 특성을 갖는 입력측 트랜스듀서를 선형화시키는 회로라 할 수 있다.

**그림** 12(a)와 같이 입력의 신호가 포물선의 형태로 나타나면 리니어라이저(Linearizer)의 회로전달함수를 어느 특정값에 맞추어 그 결과로 직선적인 형태의 출력신호를 얻는 것이다. 이때 이러한 회로 특성을 위해 승산기나 로그앰프와 같은 비선형 특성을 갖는 증폭기를 사용한다. 그림 12(b)는 다이오드를 사용한 절선근사적인 방법으로 리니어라이저를 구성한 경우이다. 이러한 회로의 경우는 입력전압이 증가하는 경우 각 다이오드에서 정해진 바이어스점을 넘게 되면 도통하게 되어 입력전압의 곡선적 형태를 압축하는 형식으로 나타나게 된다.

## 3. 전치 증폭부

전치 증폭부는 실제 ADC를 어느 곳에 응용을 하는가에 따라 대폭적으로 변화하며 이것 스스로도 여러가지 부분으로 나뉘어질 수 있다.

당장 생각해 보면 다점계측의 경우 멀티 플렉서를 떠올릴 수 있으며, 노이즈성분을 크게 함유한 경우 적당한 필터회로 또 정, 부 판정을 위한 회로 등을 생각하여야만 한다.

일반적으로 프리앰프 즉 전치 증폭기는 다음과 같은 특성이 요구된다.

① 낮은 드리프트
② 낮은 잡음성분
③ 높은 이득

이러한 요구조건을 완전히 만족하는 전치 증폭기의 설계는 쉽지 않으며 많은 기술이 요구되는데 일반적으로 차동앰프를 이상적 앰프로 간주하여 이를 사용한다.

필자는 언젠가 32점의 온도계를 설치하여 이를 ADC로 읽는 회로의 설계를 한적이 있었다. 그때 열전쌍의 출력이 mV단위의 극소 레벨이어서 신호의 입력에 애를 먹은적이 있었는데 결국 필터와 입력 바이어스전압이 0V에 가까운 바이페트(Bi-FET)로써 실현할 수 있었다. 이러한 경우 동일전압을 입력에 설치하여 반복적으로 읽어 들일때 리드 데이터가 모두 일치하지 않은 경우가 있는데, 이러한 것도 어떤 의미에서는 전치 증폭단의 드리프트와 같은 문제나 다점계측에서 소홀이될 수 있는 전원변동에 의한 요인이라 생각할 수 있다. 또한 미소신호의 측정에 있어서는 아날로그 멀티 플렉서의 문제도 배제할 수 없다. 일반적으로 아날로그 멀티 플렉서는 디지탈 멀티 플렉서와 같이 데이터 셀렉터로써 많은 신호라인중 특성신호만을 연결한 형태로 해줄 수 있는 것을 말한다.

이러한 멀티 플렉서는 J-FET나 MOSFET를 사용하는 경우가 많은데 이때 제작상 큰 문제가 되는 것이 ON시의

저항과 누설에 의한 전류이다. 특히 미소신호인 경우 누설 전류에 의한 드롭과 신호에 의한 드롭이 구분되지 않아 설계상 문제가 된다. 이러한 문제를 피하기 위해 릴레이와 같은 기계적 스위치를 사용하는 방법이 있으나 이러한 경우는 스위치 부근의 열에 의한 기전력 등에도 주의를 기울여야 한다. 경제적인 여유가 어느정도 허락되면 각 측정점마다 적당한 전치 증폭부를 설치하여 실제적인 S/N(신호대 잡음비)를 증가하여 읽는 방법이 있다. 일반적으로 고성능 ADC의 경우 각 점마다 전치 증폭부를 두는 것이 보통이다. 아날로그 멀티 플렉서의 설명에서 신호대 잡음비를 증가시키기 위해 멀티 플렉서 전단의 계기증폭에 관하여 언급하였는데 이 신호대 잡음비를 증가시키기 위한 또하나의 중요한 것이 필터링이다.

전치 증폭부의 조건을 만족하는 것으로서 차동앰프를 사용했는데 이 차동앰프의 특성의 지표로써 CMRR(Common Mode Rejection Ration)이라는 동상신호 제거비라는 것이 있는데 이는 글자 그대로 차동증폭기의 두 입력단자에 출현하는 노이즈는 통계적으로 같다고 보아 노이즈 제거에 유효한 수단이 될 수 있다.

그러나 차동앰프의 실제적 경우 입력 주파수가 커지면 커질수록 CMRR은 작아지게 되어 노이즈 문제가 다시 발생한다. 이와 같은 문제를 해결하기 위해 필터를 통해 대역별로 노이즈의 분포대역을 감쇄하여 전체적인 신호대 잡음비를 증가시킨다. 실제로 필터를 제작할 때 부품의 선정과 설계를 통해 기판에 제작하는 경우 부품의 레이아웃 등도 안정된 설계에 큰 요인이 되는데 이는 많은 경험을 필요로 한다.

일반적으로 캐퍼시터의 경우 누설전류가 작은 폴리스티롤, 마이카, 마일러 등이 사용되는데 특히 폴리스티롤 계통의 것은 고주파 특성이 우수하다. 하지만 폴리스티롤은 열에 약하다는 단점을 갖고 있다. 어떠한 필터의 설계를 하는 경우 큰 시정수가 필요한 경우가 있는데 이때는 캐퍼시터를 먼저 결정하고 저항을 결정하는 것이 좋다. 왜냐하면 위에 언급된 캐퍼시터는 용량의 크기에 제한이 있기 때문이다.

## 실제적 동작을 위하여

지금까지 ADC의 종류와 특성과 구성을 위해 필요한 부수적 회로의 종류와 의미를 설명하였다. 하지만 이러한 것 이외에도 컴퓨터와 ADC와의 인터페이스기법, 입력된 데이터의 해석요령 등의 문제가 남는다. 이러한 것은 다음 기회로 미루고 실제로 ADC를 구입하여 컴퓨터에 인터페이스 하기 위해 우선적으로 고려돼야 하는 파라미터를 소개한다. 일반적으로 이러한 파라미터의 결정에 의해 브랜드를 채택하는데 다음과 같은 파라미터가 고려대상이 된다.

1. 변환 방식
2. 변환 속도
3. 비트수
4. 분해능
5. 출력코드
6. 입력의 종류
7. 입력크기의 범위

### ●각 ADC의 비교

이미 앞에서 각각의 ADC의 변환방식은 설명이 됐으므로 이제는 변환방식에 따른 장단점을 살펴보자.

램프형 ADC의 경우 가격이 싸고 변환방식이 간단한데 반해서 온도에 의한 캐퍼시터와 저항의 변화에 따른 변화 때문에 정밀한 변환이 어렵다. 이중적분형의 경우 온도에 의한 캐퍼시터나 저항의 변화를 받지않고 클럭의 정밀도에 영향을 받지 않으므로 많이 사용되지만 변환속도가 떨어진다는 단점이 있다. 그러나 입력의 경우 변환기간내에서 신호가 변동하더라도 그 평균치가 입력되는 구조이므로 입력정보의 변동이 매우 심하거나 잡음에 대한 마진이 커질 수 있는 장점이 있다.

병렬형의 콘버터는 PCM의 복조 등에 사용되는데 주로 고속변환이 요청되는 곳에 사용된다. 하지만 고속변환이란 장점을 가지고 있는 반면 고가에 많은 비트수, 즉 분해능을 크게 할 수 없는 단점이 있다.

변환속도의 경우 용도에 의해 선정되어야 하는데 디지탈 볼트미터 등과 같이 변환속도가 느려도 되는 경우는 이중적분형의 콘버터를 사용할 수 있으며 이러한 경우 비교적 높은 정확도와 높은 분해능을 얻을 수 있다. 그러나 PCM의 복조나 화상처리 등에서는 1$\mu$s 이하의 변환시간이 요구되므로 고속변환기인 병렬형이 적합하다.

시판되는 계수형 변환기와 비교형 변환기를 비교해 보면 수ms 이하의 경우 비교적 저속에 속하므로 이 경우는 계수형이 사용되고 수백ns에서 수백$\mu$s에서는 비교방식중의 축차비교형이나 비교방식이 사용된다. 특히 100ns 이상의 고속영역에서는 비교방식중 병렬비교형이나 직병렬비교형의 것이 많이 사용된다.

이제 비트 수에 관해 생각해 보면 이는 곧 분해능과 관계된다. 이때 접속되는 컴퓨터의 프로세서 비트 수와는 아무런 관계가 없으며 다만 동일 비트를 채택할 경우 인터페이스가 간편해질 수 있다. 분해능과의 관계는 출력비트에 관계되며 분해능이란 2개의 아주 접근된 값을 얼마나 깊은 정도로 구별할 수 있는가에 대한 지표이다.

아날로그 디지탈 변환기의 분해능은 ADC가 검출할 수 있는 최소값을 얼마나 작게할 수 있는가에 따르는데 어떠한 ADC라도 디지탈 출력의 LSB를 변화 시키는 단위전압의 값이 된다. 여기서 LSB는 Least Significant Bit를 말한다. 예를 들어 8비트의 ADC의 경우 입력이 0V에서 10V까지 변한다고 가정하고 이때의 2진출력을 살펴보면 출력이 0V일 경우 $00_H$이고 10V일 경우는 $FF_H$이다.

이것은 전체 입력전압을 스텝수, 즉 $2^8$인 256으로 나눈값 0.039V가 1LSB의 변화출력이 된다.

이와 같이 이상적인 양에서는 당연히 입력전압과 차이가 발생하며 이것이 오차로써 표시되는데 이것을 양자화 오차라고 한다. 따라서 분해능을 높이면, 즉 비트 수를 늘리면 스텝의 수가 늘어 나므로 정밀도를 개선할 수 있다. 그러나 분해능을 높이는 것이 반드시 정밀도를 높인다고 말할 수는 없으며 정밀도는 분해능외의 많은 요인에 의해 결정된다. 실제적인 분해능의 선정은 정보이론에 의해 최소 분해능을 선택하지만 여기서는 실제적인 예로써 이의 설명을 대신한다.

만일 ADC의 출력을 1%이내의 오차범위내에서 작동시킨다고 하면 1%는 1/100이므로 최소한 1/100 이하의 스텝 수가 필요하므로 최소한 7비트가 필요하며 여기에 부호비트를 첨가하여 전체적으로 8비트가 필요하다.

중요한 사실은 오차범위를 설정한 1%란 값이 정밀도에 대한 오차가 아니라는 사실이다. 즉 이 1%는 콘버터에 허용되는 최대 입력전압에 대한 오차라는 사실이다. 따라서 입력최대전압이 10V일 경우 여기의 1%=0.1V는 만족할 수 있는 분해능이 될 수 있으나 입력신호가 1V일 경우는 정밀도는 10%밖에 얻을 수 없다. 이것은 앞에서 말한 분해능과 정밀도의 관계에 대한 예가 될 수도 있다.

이와 같이 분해능의 선정은 입력전압값의 범위에 따라 선택해야 하기 때문에 ADC의 다이나믹 레인지와의 관련에 유의하여야 한다.

다음은 출력코드에 대한 선정이다. 출력코드의 선정은 실제로 전산기가 많은 양의 데이터를 처리해야 되므로 시스템의 특성과 부합되는 코드를 사용해야 한다.

각 코드의 종류와 예를 살펴보면 다음과 같다.

1. 스트레이트 바이너리코드
2. 콤플리멘터리 바이너리코드
3. 오프셋 바이너리코드
4. BCD코드

스트레이트 바이너리코드는 의미대로 FS(Full Scale)에 대한 스텝의 비를 전체의 스텝에 1:1대응을 시켜놓은 형이다. 표 1과 같이 전체의 스텝과 스케일을 표시할 수 있다.

콤플리멘터리 바이너리코드는 스트레이트 바이너리코드에 대한 보수적 출력이므로 위의 값 즉 2진출력의 보수로 생각하면 된다. 그런데 스트레이트 코드나 콤플리멘터리 코드는 단일 극성 표시밖에 할 수 없기 때문에 부극성과 정극성을 모두 표현하기 위해서는 다음과 같은 방법을 사용한다.

1. 부호 절대값 표시
2. 부호화된 1의 보수표시
3. 부호화된 2의 보수표시

여기서 1의 보수로서 표시되는 경우 +0과 −0이 동시에 존재하게 되어 많이 사용되지 않고 2의 보수체계를 사용한다. 2의 보수체계를 사용하는 경우 0은 +0과 −0의 구별이 없으며 마이너스 방향으로 1스케일이 늘어나게 된다.

오프셋 바이너리코드는 결과적으로는 2의 보수형 코드와 동일하다. 즉 MSB의 비트가 사인비트로 작용하여 MSB가 1이면 플러스 그렇지 않으면 −의 구조로 된다. 표 2에 오프셋 바이너리코드의 예를 보였다.

BCD코드는 사람이 판독하기가 가장 편리한 구조로 되어 있으며 다만 사용되는 비트 수에 비하여 스트레이트 바이너리 사용시 보다 훨씬 작은 분해능을 갖는다는 단점이 있으나 사람과의 인터페이스가 쉬운 관계로 비교적 널리 사용되는 방법이다. 표 3에 이의 예를 보였다.

## 컴퓨터와의 인터페이스

앞에서 설명한 바와 같이 실제로 컴퓨터와 주변인터페이스를 결합할 경우 ADC의 선정에 있어서도 고려할 많은 사항이 있음을 지적했다. 그러면 이제 충분한 고려를 통하여 ADC의 종류를

〔표 1〕 스트레이트 바이너리코드

| MSB | BIT3 | BIT2 | LSB | 전 압 |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 9.375(FS-1LSB) |
| 1 | 1 | 1 | 0 | 8.750 |
| 1 | 1 | 0 | 1 | 8.125 |
| 1 | 1 | 0 | 0 | 7.500(3/4FS) |
| 1 | 0 | 1 | 1 | 6.875 |
| 1 | 0 | 1 | 0 | 6.250 |
| 1 | 0 | 0 | 1 | 5.625 |
| 1 | 0 | 0 | 0 | 5.000(1/2FS) |
| 0 | 1 | 1 | 1 | 4.375 |
| 0 | 1 | 1 | 0 | 3.750 |
| 0 | 1 | 0 | 1 | 3.125 |
| 0 | 1 | 0 | 0 | 2.500(1/4FS) |
| 0 | 0 | 1 | 1 | 1.875 |
| 0 | 0 | 1 | 0 | 1.250 |
| 0 | 0 | 0 | 1 | 0.625(1LSB) |
| 0 | 0 | 0 | 0 | 0.000 |

*예를 들어 1001의 경우 $V=V_{ref}\times\left(\frac{8}{16}+\frac{1}{16}\right)=5.625V$ 이다. 여기서 $V_{ref}=10V$ 이다

FS는 full scale의 약어이다

〔표 2〕 오프셋 바이너리코드

| MSB | BIT3 | BIT2 | LSB | 전 압 |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | +4.375(FS-1LSB) |
| 1 | 1 | 1 | 0 | +3.750 |
| 1 | 1 | 0 | 1 | +3.125 |
| 1 | 1 | 0 | 0 | +2.500(1/2FS) |
| 1 | 0 | 1 | 1 | +1.875 |
| 1 | 0 | 1 | 0 | +1.250 |
| 1 | 0 | 0 | 1 | +0.625 |
| 1 | 0 | 0 | 0 | 0.000(0V) |
| 0 | 1 | 1 | 1 | −0.625 |
| 0 | 1 | 1 | 0 | −1.250 |
| 0 | 1 | 0 | 1 | −1.875 |
| 0 | 1 | 0 | 0 | −2.500(−1/2FS) |
| 0 | 0 | 1 | 1 | −3.125 |
| 0 | 0 | 1 | 0 | −3.750 |
| 0 | 0 | 0 | 1 | −4.375 |
| 0 | 0 | 0 | 0 | −5.000(−FS) |

〔표 3〕 BDC 코드

| MSD | DIGIT2 | LSD | 10진수 |
|---|---|---|---|
| 1001 | 1001 | 1001 | 999 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 1010 | 0110 | 0100 | 864 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 0111 | 0111 | 0111 | 777 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 0100 | 0000 | 0000 | 500 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 0010 | 0010 | 0010 | 222 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 0001 | 0000 | 0000 | 100 |
| 0000 | 1001 | 1001 | 99 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 0000 | 0001 | 0000 | 10 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 0000 | 0000 | 0001 | 1 |
| 0000 | 0000 | 0000 | 0 |
| $10^2$ | $10^1$ | $10^0$ | |

선정했다면 어떻게 컴퓨터와 인터페이스할 것인가를 생각해야 한다.

원리적으로 컴퓨터가 ADC의 변환된 데이터를 알기 위해서 컴퓨터의 타이밍과 ADC의 타이밍이 정확히 맞아야 한다. 일반적으로는 비동기적으로 발생하는 데이터의 경우는 I/O 포트를 거쳐서 ADC를 액세스하는 경우가 대부분이다. 특히 마이크로 프로세서에 직접 ADC를 접속하는 경우 이중적분형과 같은 저속의 ADC는 별로 적합하지 못하다. 이런 경우는 축차비교형의 중·고속에 적합한 ADC를 선정하는 것이 바람직하다.

컴퓨터와 ADC의 인터페이스 방법에는 다음과 같은 방법이 있다.

1. **프로그래밍에 의한 액세스**
2. **DMA 방식에 의한 액세스**
3. **인터럽트에 의한 액세스**

재미있는 것은 CPU와 CRT를 접속하는 경우도 위와 똑같은 방법을 사용한다는 것이다. 바꾸어 생각하면 일반적으로 프로세서가 주변의 입출력장치와 데이터를 주고받기 위해서는 위와 같은 방법의 사용이 일반적이라는 것이다. 이것은 마이크로 프로세서 입장에서는 입출력장치도 하나의 메모리로서의 역할밖에 할 수 없다는 것이다.

프로그래밍에 의한 방식중 스트로브를 사용한 경우는 우선 프로세서가 ADC의 현재의 상황을 판독하여 그 ADC가 바쁘지 않으면 변환개시 신호를 주고 그렇지 않으면 변환개시 신호를 넣지 않고 자신의 다른 일을 수행하는 기법이다. 이러한 것을 **그림 13**에 도시하였다.

이와 유사한 방법으로 폴링방식이 있는데 이는 비지(Busy)신호를 읽어 바쁘면 그것이 한가하게 될때까지 대기하는 방식이며 리소스의 극대활용이라는 측면에서는 별로 추천할만한 방법은 되지 못한다.

반면에 하드웨어적인 방법을 통해 ADC를 액세스하는 경우 DMA방식은 아예 프로세스를 멈추게 하여 ADC의 판독정보를 메모리로 곧장 옮기는 것이며 이것은 잦은 판독이 필요할 경우 시스템의 특성을 잘 살펴야 한다.

일반적으로 하드웨어에 의해 잘 사용되는 방법은 인터럽트에 의한 방법이 있는데 이런 방법은 컴퓨터의 능률을 매우 높여 준다. 이는 프로세서가 현재 어떤 일을 처리하는 도중에 ADC로 부터 변환이 끝났다라는 신호가 오면 그 데이터를 판독하여 저장하고 원래의 자기일을 다시 하는 식으로 하여 처리한다. 이상과 같은 방법은 실제로 컴퓨터와 접속 즉 인터페이스하기가 비교적 용이한 경우이다. 그러면 이제 기존의 퍼스널컴퓨터를 사용하여 선택된 ADC와의 인터페이스를 해보는 실제의 회로 구성에 대해 알아보자.

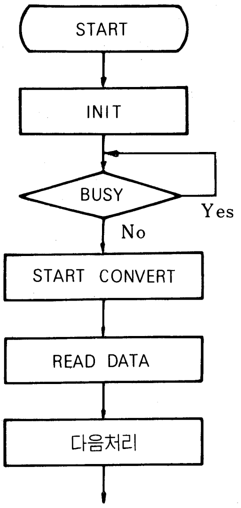
여기서는 MSX와 IBM PC에서의 인터페이스를 설명하고자 한다. 이 설명은 예제의 성격에 가까우며 앞에서 설명한 컴퓨터와의 인터페이스방법을 적용하면 어느 기종의 컴퓨터와도 쉽게 인터페이스할 수 있다.

## MSX와의 인터페이스

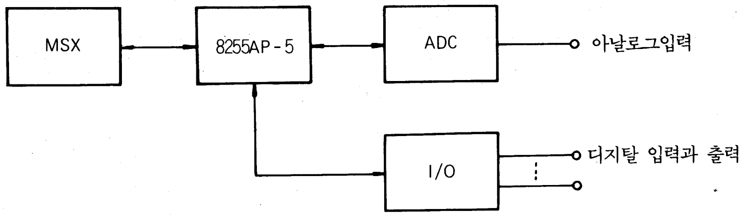
MSX컴퓨터를 사용하여 실제로 ADC를 인터페이스 하려면 Z80 마이크로 프로세서를 채택하고 있는 MSX의 CPU를 알아야 한다.

Z80은 메모리 스페이스와 입출력 스페이스가 완전히 분리되어 동작되는 형식이므로 인터페이스가 쉽게 된다.

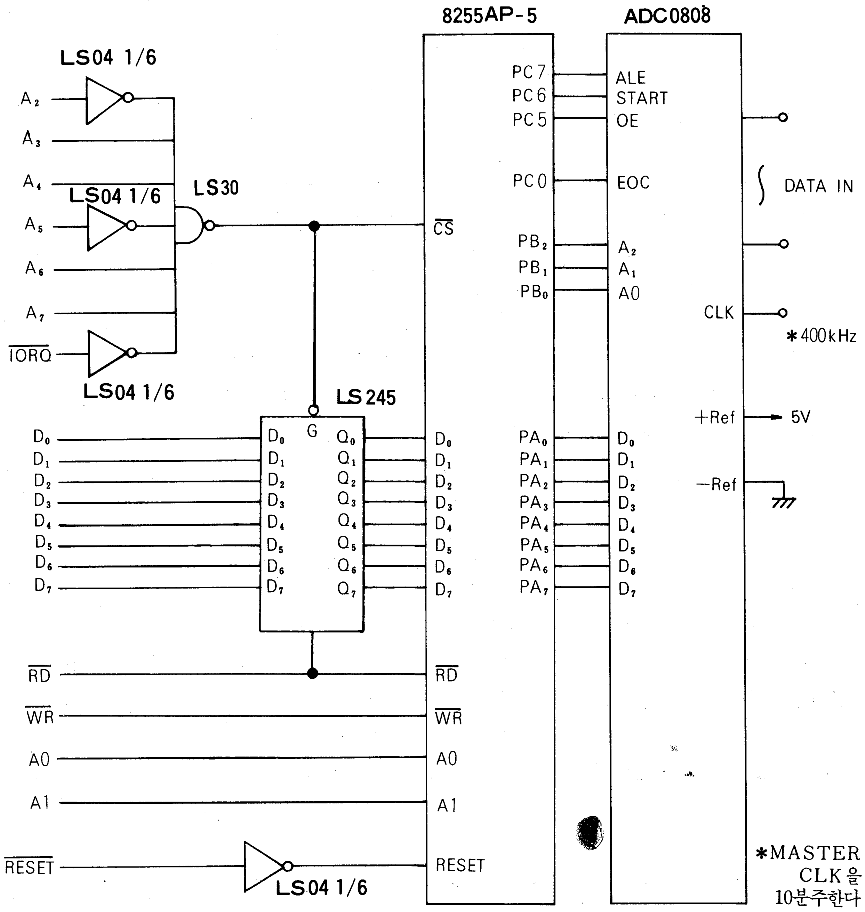
〔그림 13〕 프로그래밍에 의한 액세스방법




〔그림 14〕 MSX 접속블록도




〔그림 15〕 MSX의 접속도




그런데 MSX에 직접 ADC를 달 수도 있으나 여기서 추후에 확장하여 ADC를 이용한 데이터로 여러가지 동작을 출력하게 하기 위해서 범용 입출력장치로 많이 쓰이는 8255AP-5를 인터페이스 하고 그것으로 ADC를 물리는 형식에 대해 알아보자. 그것에 관한 블록도를 **그림 14**에 나타냈고 **그림 15**는 접속도와 프로그램 **리스트 2** 이다.

## 2. IBM PC와의 접속

IBM PC의 인터페이스 역시 MSX의 그것과 유사하다. PC에서는 $\overline{IORQ}$가 나오지 않고 $\overline{IOR}$과 $\overline{IOW}$가 나오기 때문에 오히려 인터페이스가 단순해진다.

**그림 16**은 접속도와 프로그램 **리스트 3** 이다.

지금까지 ADC의 특성과 종류 그리고 선별방법 및 실제적인 컴퓨터 접속에 까지 설명을 하였다. 실제로 활용할 경우 더욱더 많은 지식과 경험이 필요하며 특히 ADC의 입력에 관한 프리앰프부는 많은 노하우와 경험이 필요하다. 다음 호에는 읽어들인 데이터의 해석법 디지탈필터 그리고 프리앰프의 설계에 관해 논의하겠다. 끝으로 두서없는 글을 참을성있게 읽어준 분들께 감사를 드린다.

〔리스트 2〕

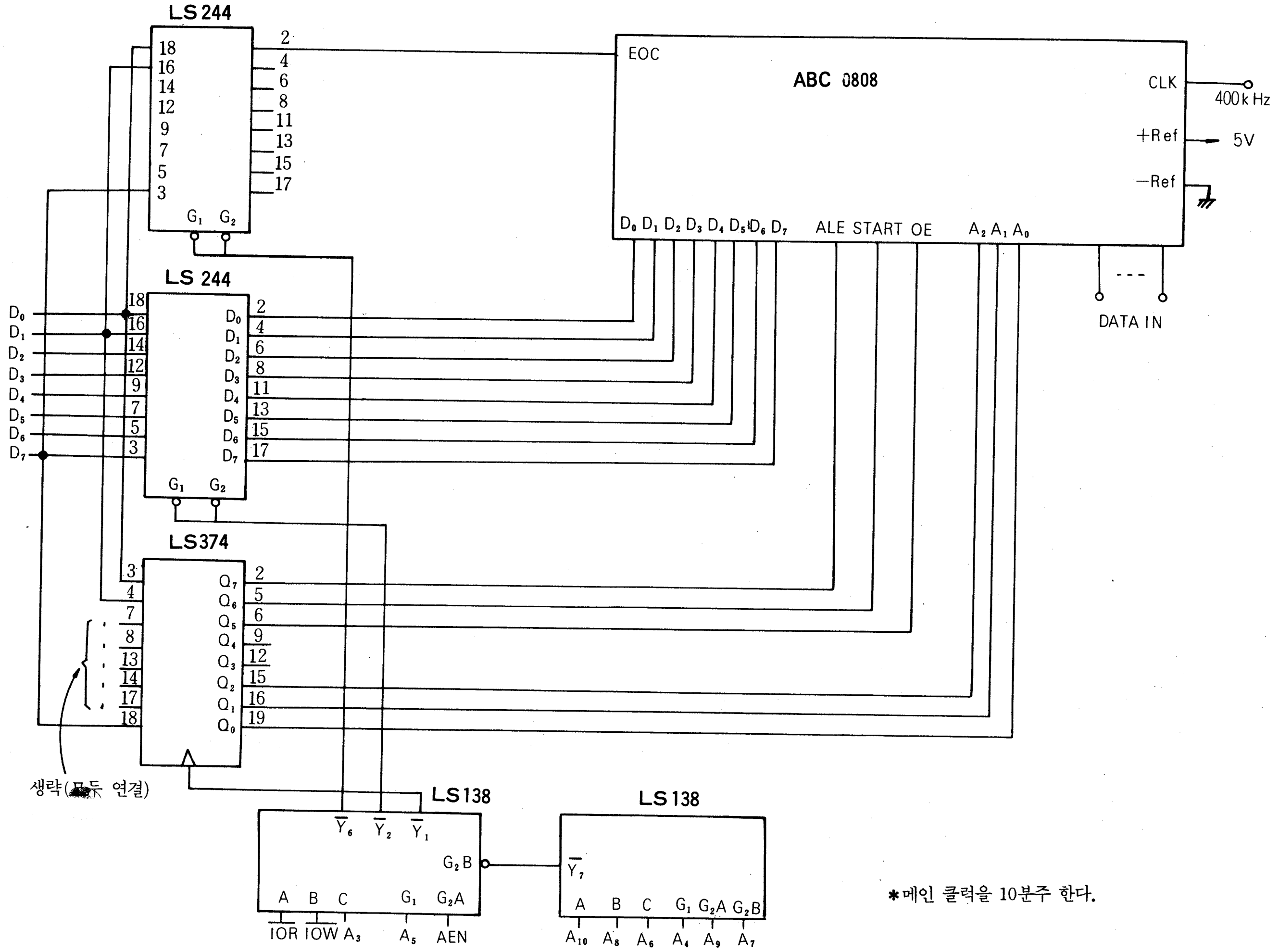
```
100 ' Successive Approximation Convertor ADC 0808 MSX Ver 1.0
110 ' Programed by C.S.W 88/10/25
120 ' SET A is IN
130 ' SET B is OUT
140 ' SET C is IN(0,3)/OUT(4,7)
150 '
160 'A=&HDS:B=&HDA:CR=&HDB
170 OUT CR,&B91
180 OUT A,0
```

```
190 OUT B,0
200 OUT C,0
210 '
220 ' ADC HANDILING PROGRAM FOR TEST
230 ' C7=ALE,C6=START,C5=OE,C0=EOC,B=ADDR,A=DATA
240 '
250 OUT B,AD           :' SET ADDR
260 OUT C,&B10000000 :' SET ALE
270 OUT C,&B11000000 :' START
280 OUT C,&B10000000
290 '
300 IF (INP(C) AND 15) THEN 310 ELSE 300 :' EOC?
310 OUT C,&B10100000 :' OE
320 PRINT INP(A)      :' IN
330 OUT C,0           :' INIT C
340 AD=AD+1           :' INC ADDR
350 IF AD=9 THEN AD=0
360 GOTO 250
```

〔리스트 3〕

```
100 ' Successive Approximation Convertor ADC 0808
110 ' IBM-PC XT Ver 1.0
120 ' Programed by C.S.W 88/10/25
130 ' ADC HANDILING PROGRAM
140 CR=0
150 AD=0
160 OUT &H580,(AD OR CR)          :' SET ADDR
170 OUT &H580,(&H80 OR AD)        :' SET ALE
180 OUT &H580,(&HC0 OR AD)        :' SET START
190 OUT &H580,(&H80 OR AD)        :'
200 IF INP(&H580) THEN 220 ELSE 210 :' EOC?
210 OUT &H580,(&H90 OR AD)        :' SET OE
220 DAT=INP(&H90 OR AD)           :' READ DATA
230 PRINT A
240 OUT &H580,0                   :' INIT CR
250 AD=AD+1                       :' INC ADDR
260 IF AD=9 THEN AD=0
270 GOTO 170
```

〔그림 16〕 IBM PC의 접속도



# 수의 부호화

김정선/항공대 전자과 교수

## 1 정보의 표현

현대 사회를 정보화사회라고한다. 정보란 우리 인간이 의사를 표현하고 전달하며 인간의 5감(五感)을 통하여 사물을 인지하는데 필요한 모든 것을 말한다고 할 수 있다. 예를 들어 양을 나타내기 위한 숫자 정보, 언어를 나타내기 위한 문자 정보, 모양을 나타내기 위한 도형 정보, 소리를 나타내기 위한 음성 정보 등을 들 수 있으며, 이밖에도 수많은 정보의 표현이 있을 수 있다. 이러한 정보를 나타내는 방법도 다양하며 각 나라마다, 각 용도마다 다를 수 있는 것이다.

예를 들어 양을 나타내기 위한 숫자 정보의 경우, 우리는 하나, 둘, 셋…으로 표현하며, 1, 2, 3…으로도 표현하고 一, 二, 三…, Ⅰ, Ⅱ, Ⅲ, …, 등 수많은 방법의 표현이 이용되고 있는 것이다. 즉, 필요에 따라 편리한 방법으로 같은 정보도 달리 표현되고 이용된다고 하겠다.

우리 인간이 필요로 하는 정보량은 대단히 많으며, 문화가 발달하면 할수록 더 많은 정보량을 필요로 한다. 이러한 정보를 표현하고 전달하며, 보관 혹은 처리하기 위하여 서로간에 약속된 어떤 몇가지의 기호 등에 의한 표현방법을 사용한다. 숫자를 나타내기 위한 1, 2, 3, …이나 문자의 표현으로 A, B, C, … 혹은 ㄱ, ㄴ, ㄷ… 등의 예를 들 수 있다.

오늘날 우리 사회는 컴퓨터사회화되면서 이러한 정보의 표현 방법도 인간들 사이에서만 편리하고 용이한 방법으로가 아니고, 기계화함에 적합한 방향으로 바뀌어가고 있다. 많은 정보량을 가능한 적은 기호(장치 혹은 기구) 만으로 능률적인 표현을 할 수 있도록 하는 것은 대량의 정보를 능률적이고 경제적으로 처리하여야 하는데 필수조건이라 할 수 있다.

숫자 정보표현에서 우리는 일반적으로 10진법의 표현, 즉 "0"이라는 기호 "1"이라는 기호…"9"라는 기호 등, 각기 다른 10가지의 기호로서 나타내는 방법을 익혀 왔으며 "수"하면 주로 10진법을 말하며, 다른 진법에 관하여는 거의 의식하지 못하고 사용해 왔다. 그러나 두 가지의 기호만을 이용하는 2진법, 3가지의 기호를 이용하는 3진법, 16가지 각기 다른 기호를 이용하는 16진법 등 수많은 진법도 생각할 수 있으며, 보다 적은 기호로서 보다 많은 정보를 나타낼 수 있는 자연수로서 가장 능률적인 진법은 3진법이다.

간단한 예로써 "0", "1", "2"… "9"와 같은 10개의 기호가 그려져 있는 카드로 표현할 수 있는 임의의 수는 10가지이다. 예를 들어 "0"기호 3장, "1"기호 3장, "2"기호 3장 즉, 9장의 카드로 조합하여 표현할 수 있는 정보량은 27가지이다.

3진법 다음으로 능률적인 진법은 2진법과 4진법을 들 수 있다.

전자계산기는 전기회로로 구성되며, 회로구성으로 진법에 대응시켜 실현함에는 2진법이 가장 유리하다. 즉, 두가지의 기호에 대응하여 전기적으로 ON/OFF, 혹은 전압의 유, 무 등에 간단히 대응시킬 수 있기 때문이다. 따라서 현재의 컴퓨터 시스템은 물론 전자회로를 이용하는 대부분의 디지탈 시스템은 2진법에 의한 정보의 표현방법이 이용되고 있다.

## 2 수의 진법과 변환

### (1) 수의 표시

R진법으로 수 N을 나타내는 일반적인 표시식은

$$N=\sum_{i=-\infty}^{\infty} d_i R^i \qquad \cdots\cdots\cdots ①$$

로 표시되며, 여기서 $d_i<R$이고, $R$은 기수(Radix)이며, $d_i$는 $R$진법에서 $i$자리의 임의의 숫자이며 $i$는 정수이다.

예를 들어 10진법으로($R=10$) 6217은

$$6217=6\times10^3+2\times10^2+1\times10^1+7\times10^0 \qquad \cdots\cdots\cdots ②$$

이며, 이때 $i=3$, $d_0=7$, $d_1=1$, $d_2=2$, $d_3=6$이다.

2진법으로 11010는

$$11010_2=1\times2^4+1\times2^3+0\times2^2+1\times2^1+0\times2^0=26_{10} \qquad \cdots\cdots\cdots ③$$

여기서 $R=2$, $i=4$, $d_0=0$, $d_1=1$, $d_2=0$, $d_3=1$, $d_4=1$이다.

2진법을 표현하는 경우 위에서와 같이 "0"와 "1"의 두가지 기호를 사용하는 것이 현재 통례이지만, 또 다른 방법으로 "+", "−", 혹은 ON, OFF 등

다양한 표현방법도 생각할 수 있다. 일반적으로 10진수 미만의 진법에서는 10진법에서 표시되는 아라비아숫자의 기호를 이용하는 경우가 많다.

예를 들어 2진법에서 "0"와 "1", 3진법에서 "0", "1", "2", 4진법에서 "0", "1", "2", "3" 등으로 표시하며, 10진법 이상에서는 10진법의 "0"~"9"의 10개 기호 외에 필요한 수만큼의 다른 기호를 사용하여 표시한다. 예를 들어, 16진수 표현의 경우 16가지의 기호중 10개는 "0"~"9"까지 10진수의 기호를 사용하며 나머지 6개는 알파벳의 "A", "B", "C", "D", "F"를 사용하는 것이 보통이다.

전술한 바와 같이 디지탈 시스템에서는 대부분 2진법의 수를 사용한다. 이러한 2진법은 능률적이고 경제적인 수 조직이기는 하지만 사람이 인식, 혹은 표현하는데는 불편한 점이 많다. 따라서 컴퓨터나 디지탈 시스템내에서는 2진법으로, 그리고 사람들 사이에서는 10진법 혹은 16진법, 8진법 등으로 변환하여 사용하는 것이 일반적이다. 2진법과 변환관계에서 10진법이나 또 다른 진법으로의 변환은 8진법 혹은 16진법으로 변환하는 것에 비교하여 복잡하다. 따라서 2진법을 주로 이용하는 디지탈 시스템 등에서는 8진법, 16진법의 표현사용이 병행되는 것이 일반적이다.

표 1은 10진법, 2진법, 8진법, 16진법의 표시방법의 예이다.

〔표 1〕 각 진법에 따른 수의 표시

| 10진법 | 2진법 | 8진법 | 16진법 | 3진법 | 10진법 | 2진법 | 8진법 | 16진법 | 3진법 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 9 | 1001 | 11 | 9 | 100 |
| 1 | 1 | 1 | 1 | 1 | 10 | 1010 | 12 | A | 101 |
| 2 | 10 | 2 | 2 | 2 | 11 | 1011 | 13 | B | 102 |
| 3 | 11 | 3 | 3 | 10 | 12 | 1100 | 14 | C | 110 |
| 4 | 100 | 4 | 4 | 11 | 13 | 1101 | 15 | D | 111 |
| 5 | 101 | 5 | 5 | 12 | 14 | 1110 | 16 | E | 112 |
| 6 | 110 | 6 | 6 | 20 | 15 | 1111 | 17 | F | 120 |
| 7 | 111 | 7 | 7 | 21 | 16 | 10000 | 20 | 10 | 121 |
| 8 | 1000 | 10 | 8 | 22 | 17 | 10001 | 21 | 11 | 222 |

## (2) 수의 변환

우리가 익숙해져 있는 수는 10진법에 의하여 표시되는 수 조직이다. 따라서 디지탈 시스템 혹은 컴퓨터 시스템에서 주로 이용되는 2진법, 8진법, 16진법으로의 상호 변환관계를 익혀두는 것이 필요하다.

먼저 10진수를 임의의 진수로 변환하는 경우, 기수 10인 임의의 수 $N$을 기수 $R$인 $R$진법으로의 변환은 $N$을 $R$로 나누고 나머지를 $d_0$, 몫을 $N_1$, 다시 $N_1$을 $R$로 나누고, 나머지를 $d_1$, 몫을 $N_2$ …로 반복하여

$$R_N = d_nR^n + d_{n-1}R^{n-1} \cdots + d_1R^1 + d_0R^0 \quad \cdots\cdots\cdots ④$$

로 표시한다.

예를 들어 10진수 30($N$=30)을 2진법($R$=2)으로 변환하면

2) 30
2) 15……0 ← $d_0$
2) 7……1 ← $d_1$
2) 3……1 ← $d_2$
2) 1……1 ← $d_3$
0……1 ← $d_4$

따라서

$$R_n = 11110 \quad (d_4\, d_3\, d_2\, d_1\, d_0) \quad \cdots\cdots\cdots ⑤$$

앞의 설명은 소수점 이상의 수를 변환하는 경우이며, 소수점 이하의 수는 소수점 이하의 10진수가 $N'$라면 $R'$진법으로 변환하는 경우 $N'$에 $R'$를 곱하여 소수점 위의 수를 $d'$, 다시 소수점 이하의 수를 $N_1'$라 할 때, $N_1'$에 $R'$를 곱하여 소수점 위의 수를 $d_2'$, 소수점 이하의 수를 $N_2'$로 하여 같은 계산을 반복하였을 경우, 그 결과는

$$R_N' = d_1R^{-1} + d_2R^{-2} + \cdots d_nR^{-n} \cdots ⑥$$

으로 된다. 예를 들어 $0.6875_{10}$를 2진수로 표현하면

$N'$ = 0.6875, $R'$ = 2, $d_1'$ = 1, $N_1'$ = .3750
0.6875 × 2 → 1.3750
0.375 × 2 → 0.750 ($d_2'$ = 0, $N_2'$ = .750)
0.750 × 2 → 1.50 ($d_3'$ = 1, $N_3'$ = .50)
0.50 × 2 → 1.0 ($d_4'$ = 1, $N_4'$ = .0)
0.0 × 2 → 0

따라서

$$R_n' = 0.1011 \quad (d_1'\, d_2'\, d_3'\, d_4')$$

와 같이 된다.

수의 변환에서 10진수를 2진수, 8진수 혹은 16진수로의 변환은 디지탈 시스템을 하는데 빈번히 이용되며, 따라서 이러한 변환을 능숙하고 신속하게 할 수 있도록 하는 것이 필요하다. 앞에서 설명한 방법은 일반적으로 임의의 진법으로 변환하는 방법으로 2진법 등으로의 변환에서 실제로 활용하는 좀 더 편리한 방법은 $2^0, 2^1, 2^2$ …에 해당하는 10진수의 1, 2, 4, 8, 16…를 기억하여 임의의 10진수 $N$을 2진수로 변환하는 경우, $N \geq 2^i$ 되는 $2^i$값을 $N$에서 빼고 나머지를 $N_1$이라 하며, 다시 $N_1 \geq 2^{i_1}$이 되는 $2^i$값을 $N_1$에서 빼는 방법을 반복하여 변환하면 편리하다.

예를 들어 $34304_{10}$을 2진수로 변환하는 경우 앞의 방법으로 계산하려면, 16번 2로 나누는 계산을 하여야 하나, 후자의 방식으로 하면

$34304 \geq 32768\,(2^{15})$ 이므로

**34304 − 32768 = 1536**

$1536 \geq 1024\,(2^{10})$ 이므로

**1536−1024=512**

512 ≥512($2^9$)이므로

**512−512=0**

따라서

$R_N = 1000011000000000$

$2^{15}$　$2^{10}$　$2^9$

와 같이 간단히 계산할 수 있다.

8진법, 16진법으로의 변환은 2진법으로의 변환된 값으로부터 좀 더 용이하게 구할 수 있다. 즉, 2진표현 3자리는 8진법에, 2진표현 4자리는 16진법에 대응되므로 10진수를 2진수로 변환하고, 이를 다시 8진 또는 16진수로 변환하는 것이 편리하다. 예를 들어 2진수 11001001101을 8진, 16진수로 표시하면

**11001001101** ······ 8진수변환
3 1 1 5

**11001001101** ······16진수변환
6 4 D

과 같이 된다.

이와 같이 2진수와 8진 혹은 16진수의 변환이 용이하므로 디지탈 시스템에서 2진법과 동시에 16진법, 8진법을 많이 사용하며 2진법은 시스템의 기구적 실현에서 그리고 8진, 16진법은 사람들 사이에서의 표현방법으로 주로 이용된다.

8진 또는 16진수를 2진수로 변환하는 것도 역시 다른 진법으로 변환하는 것에 비교하여 용이하며, 2진수를 10진수로의 변환은 ①식과 ③식에서 표시하는 바에 따라 변환이 됨을 알 수 있다. 예를 들어 $6AC3_{16}$을 10진수로 변환하는 경우 10진수 $N$은

16진수 **6** **A** **C** **3**
↓ ↓ ↓ ↓
2진수→110 1010 11000011

→2진수 **110101011000011**
$2^{14}$ $2^{13}$ $2^{11}$ $2^9$ $2^7$ $2^6$ $2^1$ $2^0$

이므로

$$N = 2^{14}+2^{13}+2^{11}+2^9+2^7+2^6+2^1+2^0$$
$$= 16384+8192+2048+512+128+64+2+1=27331$$

이 된다.

## 3 2진부호

사람이 필요로 하는 정보의 표현방법은 매우 다양하며, 컴퓨터나 디지탈 시스템에서 유용하게 이용되는 한 방법이 2진부호라 할 수 있다. 2진수 한자리는 0 또는 1로 2가지의 정보표현이 가능하다. 2진수 2자리는 각 자리에서 2가지씩 총 4가지의 정보표현이 가능하다. 3자리의 경우 각 자리의 0과 1의 조합은 각각 다른 8가지 방법이 가능하다. 즉, 8가지의 다른 정보의 표현이 가능하다. 2진수 한자리를 한 bit(Binary digit의 약어)라고 한다.

$n$-비트의 2진표현은 $2^n$가지의 정보를 표현할 수 있다.

2진 부호(Binary Code)는 0, 혹은 1로 표시되는 일정수의 비트로서 표시하고자 하는 정보에 대응시킨 것을 말한다. 예를 들어 10진수 "0"이라는 기호(Zero 정보)를 2진 부호 "0000"으로, "1"을(하나라는 정보) "0001"로, 혹은 알파벳 "A"라는 기호를 2진부호 "01000001"으로 표시하는 것 등을 들 수 있다. 이와 같이 2진 부호화 하는 것은 디지탈 시스템에서 전자회로의 ON, OFF(혹은 전압의 유, 무)에 쉽게 대응이 되는 것이 2진법이기 때문이다.

### (1) BCD코드

10진수는 0~9까지의 서로 다른 기호(정보)의 수 표현방법이며, 이를 2진 부호로 충분히 표현하기 위하여는 4비트가 필요하다. 역으로 4비트의 2진 부호는 최대 16가지의 정보를 표현하는 것이 가능하다. 따라서 10진수를 표현하는데는 4비트로 충분하다. 2진수 4비트로 구성되는 10진수의 표현방법은 약 2.9×$10^{10}$가지 방법이 있다.

표 2는 10진수의 4비트 2진 부호 표현방법의 몇가지 예이다.

표 1의 예에서 각 방법은 0~9까지 10진수를 나타내는 4비트의 0와 1의 조합으로 각각 다르며, 이와 같이 4비트의 조합으로 10개의 각각 다른 10진수에 대응되는 부호를 만드는 방법이 상기한 바와 같이 많이 존재하는 것이다. 이와 같이 많은 방법중에서 4가지를 표에 예시하였으며, 특히 사용하기 편리한 혹은 특수한 부호 구성의 몇가지 방법이 아주 유효하게 이용되고 있다.

표 2에서 1의 방법은 0~9까지 10진수를 2진조합으로 표현하는데 2진계수방법과 동일하다. 예를 들어 0101은 2진수로 5에 해당되며 1000는 8에 해당한다. 즉, 2진 부호와 2진수가 일치하는 방법으로 10진수 0~9까지를 표시하는 방법이며, 이 부호방식

〔표 2〕 4비트 2진 부호의 BCD코드

| 10진수 | 4비트 2진 부호 | | | |
|---|---|---|---|---|
| | 1의 방법 | 2의 방법 | 3의 방법 | 4의 방법 |
| 0 | 0000 | 0001 | 1111 | 0011 |
| 1 | 0001 | 0010 | 0000 | 0100 |
| 2 | 0010 | 0011 | 0010 | 0101 |
| 3 | 0011 | 0100 | 0001 | 0110 |
| 4 | 0100 | 0101 | 1110 | 0111 |
| 5 | 0101 | 0110 | 1100 | 1000 |
| 6 | 0110 | 0111 | 1001 | 1001 |
| 7 | 0111 | 1000 | 1010 | 1010 |
| 8 | 1000 | 1001 | 0011 | 1011 |
| 9 | 1001 | 1010 | 0100 | 1100 |

을 8421코드라 한다.

이 코드는 4비트중 가장 왼쪽 비트에 8, 다음 비트에 4, 다음 비트에 2 그리고 가장 우측비트에 1을 곱하여 더한 값이 바로 10진수가 되도록 구성한 코드이며, 이와 같이 각 비트 위치에 일정한 수를 곱하여 더하므로서 10진수의 값이 되도록 하는 코드를 가중치(Weight)를 가지는 코드라고 한다.

8421코드로서 $594_{10}$은

```
 5     9     4
 ↓     ↓     ↓
0101  1001  0100
```

로 표시되며 이 수를 2진수로 나타내면

```
 594
-512   — 2⁹
  82
 -64   — 2⁶
  18
 -16   — 2⁴
   2
  -2   — 2¹
   0
```

이므로

```
1001010010
↑  ↑ ↑  ↑
2⁹ 2⁶ 2⁴ 2¹
```

이 된다.

8421코드는 10진수를 2진화 10진수(Binary Coded Decimal Number : BCD코드)로 변환하기 용이한 장점이 있으나, 일반적인 2진수 연산이 불가능하다.

8421코드에 $(3)_{10}=(0011)_2$을 각각 더하여 얻어지는 코드를 3초과 코드(Excess three code, 표 2의 4의 방법)라고 한다. 이 코드는 8421 코드에 비하여 연산처리에서 유리한 점을 갖고 있다.

BCD코드중 유용하게 사용되는 코드의 몇가지 예를 들면 **표 3**과 같다.

이와 같은 BCD코드는 가중치(Weight)가 주어진 웨이티드(Weited) 코드, 가중치가 주어지지 않은 언웨이티드코드, 대칭성에 따라 대칭코드와 비대칭코드 등으로 분류되며 웨이티드코드는 10진수로부터 2진코드로 혹은 그 역으로의 변환이 용이한 이점을 갖고 있으며, 대칭코드는 연산처리(주로 2진 가산)가 유리한 장점을 갖는다.

## (2) 영문숫자코드

4비트의 조합으로는 16가지의 정보를 나타낼 수 있으므로 10진수를 표현하는데는 충분하나 그 외의 문자정보, 즉 알파벳이나 한글의 자모음 혹은 특수기호들을 나타내기에는 충분하지 않다. 숫자와 알파벳, 그리고 특수기호(예로 . , , , ?, ; 등)들의 보통 사용되는 수는 대략 60개 정도이다. 즉, 0~9까지 10개, A~Z까지 알파벳 대문자 24자, 소문자 24자와 몇개의 기호를 합하여 64문자로 보통 영문 문장 표현이 가능하며 이러한 영문숫자를 나타내기에 충분한 2진비트수는 6($2^6$=64)이 된다. 그러나 6비트의 6 이란 숫자는 $2^n$에 맞지 않는 수이므로, 2진법을 주로 사용하는 디지탈 시스템계에서는 적합하지 않은 점이 있다. 따라서 $2^3$에 해당하는 8비트 코드를 구성할 경우 좀더 많은 정보의 표현($2^8$=256)이 가능하며 $2^n$표현 역시 $n$=3으로 편리하게 된다.

숫자와 문자 특수기호들을 총 망라하여 표현하는 방법으로는 위와 같은 이유로 8비트의 부호구성방법을 주로 이용하고 있으며, 대표적인 2진 8비트 코드는 ASCII(American Standard Code for Information Interchange)와 EBCDIC(Extended BCD Interchange Code)를 들 수 있으며, 그 예는 **표 4**와 같다.

## (3) 패리티비트코드

2진수를 사용하는 디지탈 시스템에서 연속적으로 데이터를 이동하는 경우에 1과 0의 오류(Error)가 생기는 경우가 있다. 이와 같은 오류검출에 사용하는 방법중의 하나는 정보데이터에 여분의 한비트 또는 그 이상의 비트를 추가하여 코드내의 1의 수가 홀수 또는 짝수가 되도록 하여 오류를 검출하는 것이 있다. 이때 정보데이터에 한비트를 추가하는 비트를 패리티비트라고 한다. 코드내의 1의 수가 홀수이면 홀수 패리티(Odd parity), 1의 수가 짝수이면 짝수 패리티(Even parity)라고 한다. 이 패리티비트는 보통 맨 우측에 추가시키는데 **표 5**에 BCD코드에 패리티비트를 추가한 5비트군의 코드를 표시하였다.

따라서 짝수 패리티를 만들려면 한 코드에서 1의 수를 세어서 짝수이면 0을, 홀수이면 1을 추가하여 1의 갯수가 짝수가 되도록 하고, 홀수 패리티를 만들려면 반대로 1의 수가 홀수이면 0, 짝수이면 1을 추가하여 1의 총수가 홀수가 되도록 한다.

이렇게 하여 구성된 코드에서 1의 갯수를 검사하는 것을 패리티검사(Parity check)라고 하는데, 이 패리티검사는 특히 자기테이프, 디스크, 자심, 종이테이프와 같은 기억장치에서 흔히 사용한다. 자기테이프는 다른 것에 비

〔표 3〕 각종 4비트 코드

| 10진수 | BCD 8421 | 3초과코드 | 그레이코드 | 2421 | 5421 | 7421 |
|---|---|---|---|---|---|---|
| 0 | 0000 | 0011 | 0000 | 0000 | 0000 | 0000 |
| 1 | 0001 | 0100 | 0001 | 0001 | 0001 | 0001 |
| 2 | 0010 | 0101 | 0011 | 0010 | 0010 | 0010 |
| 3 | 0011 | 0110 | 0010 | 0011 | 0011 | 0011 |
| 4 | 0100 | 0111 | 0110 | 0100 | 0100 | 0100 |
| 5 | 0101 | 1000 | 0111 | 1011 | 1000 | 0101 |
| 6 | 0110 | 1001 | 0101 | 1100 | 1001 | 0110 |
| 7 | 0111 | 1010 | 0100 | 1101 | 1010 | 1000 |
| 8 | 1000 | 1011 | 1100 | 1110 | 1011 | 1001 |
| 9 | 1001 | 1100 | 1111 | 1111 | 1100 | 1010 |

〔표 4〕 각종 형식의 영문숫자 코드

| 문자 | 6비트 내부코드 | 7비트 ASCII코드 | 8비트 EBCDIC코드 | 12비트 카드코드 |
|---|---|---|---|---|
| A | 010 001 | 100 0001 | 1100 0001 | 12, 1 |
| B | 010 010 | 100 0010 | 1100 0010 | 12, 2 |
| C | 010 011 | 100 0011 | 1100 0011 | 12, 3 |
| D | 010 100 | 100 0100 | 1100 0100 | 12, 4 |
| E | 010 101 | 100 0101 | 1100 0101 | 12, 5 |
| F | 010 110 | 100 0110 | 1100 0110 | 12, 6 |
| G | 010 111 | 100 0111 | 1100 1111 | 12, 7 |
| H | 011 000 | 100 1000 | 1100 1000 | 12, 8 |
| I | 011 001 | 100 1001 | 1100 0001 | 12, 9 |
| J | 100 001 | 100 1010 | 1101 0001 | 11, 1 |
| K | 100 010 | 100 1011 | 1101 0010 | 11, 2 |
| L | 100 011 | 100 1100 | 1101 0011 | 11, 3 |
| M | 100 100 | 100 1101 | 1101 0100 | 11, 4 |
| N | 100 101 | 100 1110 | 1101 0101 | 11, 5 |
| O | 100 110 | 100 1111 | 1101 0110 | 11, 6 |
| P | 100 111 | 101 0000 | 1101 0111 | 11, 7 |
| Q | 101 000 | 101 0001 | 1101 1000 | 11, 8 |
| R | 101 001 | 101 0010 | 1101 1001 | 11, 9 |
| S | 110 010 | 101 0011 | 1110 0010 | 0, 2 |
| T | 110 011 | 101 0100 | 1110 0011 | 0, 3 |
| U | 110 100 | 101 0101 | 1110 0100 | 0, 4 |
| V | 110 101 | 101 0110 | 1110 0101 | 0, 5 |
| W | 110 110 | 101 0111 | 1110 0110 | 0, 6 |
| X | 110 111 | 101 1000 | 1110 0111 | 0, 7 |
| Y | 111 000 | 101 1001 | 1110 1000 | 0, 8 |
| Z | 111 001 | 101 1010 | 1110 1001 | 0, 9 |
| 0 | 000 000 | 011 0000 | 1111 0000 | 0 |
| 1 | 000 001 | 011 0001 | 1111 0001 | 1 |
| 2 | 000 010 | 011 0010 | 1111 0010 | 2 |
| 3 | 000 011 | 011 0011 | 1111 0011 | 3 |
| 4 | 000 100 | 011 0100 | 1111 0100 | 4 |
| 5 | 000 101 | 011 0101 | 1111 0101 | 5 |
| 6 | 000 110 | 011 0110 | 1111 0110 | 6 |
| 7 | 000 111 | 011 0111 | 1111 0111 | 7 |
| 8 | 001 000 | 011 1000 | 1111 1000 | 8 |
| 9 | 001 001 | 011 1001 | 1111 1001 | 9 |
| blank | 110 000 | 010 0000 | 0100 0000 | no punch |
| . | 011 011 | 010 1110 | 0100 1011 | 12, 8, 3 |
| ( | 111 100 | 010 1000 | 0100 1101 | 12, 8, 5 |
| + | 010 000 | 010 1011 | 0100 1110 | 12, 8, 6 |
| $ | 101 011 | 010 0100 | 0101 1011 | 11, 8, 3 |
| * | 101 100 | 010 1010 | 0101 1100 | 11, 8, 4 |
| ) | 011 100 | 010 1001 | 0101 1101 | 11, 8, 5 |
| — | 100 000 | 010 1101 | 0110 0000 | 11 |
| / | 110 001 | 010 1111 | 0110 0001 | 0, 1 |
| , | 111 011 | 010 1100 | 0110 1011 | 0, 8, 3 |
| = | 001 011 | 011 1101 | 0111 1110 | 8, 6 |

해 오류가 생기기 쉬워서 2중 패리티 검사를 행하는데, **그림** 1과 같이 각 어(Word)의 우측 끝에 홀수 패리티비트를 넣고, 각 어의 맨 마지막 행 끝에 홀수 패리티어(Parity word)를 추가한다. 이때 각 열과 행의 1의 총수가 홀수가 되는데, 만약 홀수가 아니면 안되는 곳의 비트자리가 오류임을 알 수 있다.

### (4) 해밍코드

패리티비트에 의한 오류검출은 단지 검출만으로 끝나고 오류의 정정은 할 수 없다. R. W. Hamming에 의해 고안된 해밍코드(Hamming Code)는 자신이 오류를 정정할 수 있다.

예로써, 4비트 데이터가 다음과 같이 전송되었다고 하자.

$$D_7D_6D_5P_4D_3P_2P_1$$

여기서 $D$는 데이터비트, $P$는 패리티비트이다. $P_1$값은 $P_1$, $D_3$, $D_5$, $D_7$이 짝수 패리티가 되도록, $P_2$값은 $P_2$,$D_3$, $D_6$, $D_7$이 짝수 패리티가 되도록, $P_4$값도 역시 $P_4$, $D_5$, $D_6$, $D_7$이 짝수 패리티가 되도록 정해져 있다. 따라서 데이터 1011이 전송될 때 짝수 패리티 7비트 해밍코드는 다음과 같다.

$$D_7D_6D_5P_4\,D_3P_2P_1$$

**1 0 1 0 1 0 1**

일반적으로 해밍코드는 비트수에는 제한을 두지 않으나, 패리티비트 $P$의 위치는 $2^n$번째가 되도록 한다(여기서 $n=0$, 1, 2, 3…).

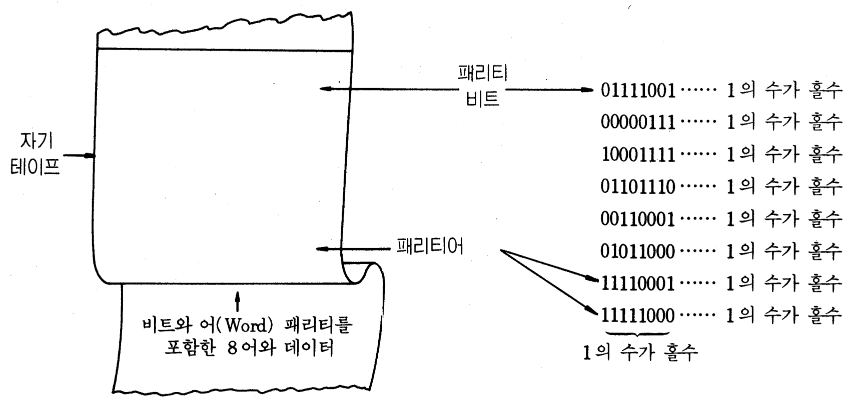
위의 예에서 해밍코드화가 된 데이터를 읽어 오류가 있는지를 검사하는데, 비트위치 1, 3, 5, 7과 2, 3, 6, 7, 그리고 4, 5, 6, 7의 1의 수가 짝수로 되는가를 조사한다. 만약 모두가 짝수 패리티로 되어 있으면 오류가 없다고 판단되고, 오류가 발견되면 몇번째 위치에 오류가 있는지를 알 수 있게 된다.

만약 해밍코드화 데이터가 1011011이면 비트위치 4, 5, 6, 7을 검사하면 오류를 발견할 수 있다(왜냐 하면, 1의 수가 홀수이므로). 이때 오류지정어의 $2^2$의 위치에 1을 기록한다. 다음에 비트위치 2, 3, 6, 7을 검사하면 짝수이므로 오류가 없어서 오류지정어

〔표 5〕 패리티 비트를 가진 BCD코드

| 10진수 | BCD코드 | 짝수 패리티 비트를 가한 BCD코드 | 홀수 패리티 비트를 가한 BCD코드 |
|---|---|---|---|
| 0 | 0000 | 0000 0 | 0000 1 |
| 1 | 0001 | 0001 1 | 0001 0 |
| 2 | 0010 | 0010 1 | 0010 0 |
| 3 | 0011 | 0011 0 | 0011 1 |
| 4 | 0100 | 0100 1 | 0100 0 |
| 5 | 0101 | 0101 0 | 0101 1 |
| 6 | 0110 | 0110 0 | 0110 1 |
| 7 | 0111 | 0111 1 | 0111 0 |
| 8 | 1000 | 1000 1 | 1000 0 |
| 9 | 1001 | 1001 0 | 1001 1 |

〔그림 1〕 자기테이프의 2중 패리티 검사




의 $2^1$ 위치에 0을 기록하고, 끝으로 비트위치 1, 3, 5, 7을 검사하면 오류가 있으므로 오류지정어의 $2^0$의 자리에 1을 기록한다. 즉, 오류지정어는 101이 되어 $D_5$가 오류가 있는 것을 알 수 있다. $D_5=1$이었으므로 $D_5=0$으로 하여 오류가 정정된다. 또 하나의 예를 들어보자.

7비트 해밍코드가 1111101인 경우 오류가 있는가를 조사해 보자.

```
D7 D6 D5 P5 D3 P2 P1                    오류지정어
1  1  1  1  1  0  1                       0 1 0
비트위치 4, 5, 6, 7에서 오류 없음 ─────────┘ │ │
비트위치 2, 3, 6, 7에서 오류 있음 ───────────┘ │
비트위치 1, 3, 5, 7에서 오류 없음 ─────────────┘
```

($D_7D_6D_5P_5D_3P_2P_1$)

오류지정어가 010이므로 비트위치 2에 오류가 있다. 따라서 1111111이 올바른 데이터이다.

## (5) 그레이코드

디지탈 시스템에서 불연속적인 데이터, 즉 이산적인 데이터를 처리하는데 대개의 물리계는 연속적인 데이터이다. 그러므로 디지탈 시스템에 사용되는 디지탈형의 데이터로 물리계의 연속적인 양을 변환해야 한다.

이와 같은 용도에 사용되는 것이 A/D변환기인데, 이 A/D변환에 사용되는 코드가 그레이코드(Gray Code, Reflected Code)이다. 이 그레이코드는 또한 주변장치에서도 많이 사용되는 코드이다. 표 6과 같이 한 수에서 다음의 수의 코드를 비교해 보면 단지 한비트만 다르다. 어느 수에서 바로 다음 수로 변환될 때 단지 한비트, 즉 0에서 1로, 또는 1에서 0으로 바뀌어진다.

표 6에서 10진수 5에서 6으로 바뀔 때 0111에서 0101로 변환되어 간다. 즉, 최하위 비트에서 두번째가 1에서 0으로 바뀐 것이다. 또 10진수 13에서 14로 될 때 그레이코드는 1011에서 1001로, 왼쪽에서 3번째자리 비트 1이 0으로 되었다. 그러므로 입력정보를 나타낼 때 오류가 적어서 입출력시스템에서 주로 사용되는 코드이다.

그러면 여기서 2진코드를 그레이코드로, 또는 반대로 변환하는 법을 알아보자. 먼저 2진코드에서 그레이코드로 변환하는 과정은 다음과 같다. 왼쪽에서 첫번째 2진자리수 MSD(most significant digit)는 그대로 그레이코드수가 된다.

```
1100…… 2진코드
↓
1    …… 그레이코드
```

다음에 오는 그레이코드수는 2진코드의 서로 인접하는 두 비트를 얻는데, 이때 +에서의 자리올림은 무시한다.

```
 +
⌒
1 1 0 0 …… 2진코드
↓ ↓
1 0     …… 그레이코드
```

〔표 6〕 4비트 그레이 코드

| 10진수 | 그레이코드 | 10진수 | 그레이코드 |
|---|---|---|---|
| 0 | 0000 | 8 | 1100 |
| 1 | 0001 | 9 | 1101 |
| 2 | 0011 | 10 | 1111 |
| 3 | 0010 | 11 | 1110 |
| 4 | 0110 | 12 | 1010 |
| 5 | 0111 | 13 | 1011 |
| 6 | 0101 | 14 | 1001 |
| 7 | 0100 | 15 | 1000 |

다음에 오는 그레이코드수도 마찬가지로 왼쪽에서 2번째 비트와 3번째 비트를 합하여 얻는다.

즉, 다음과 같다.

```
 +  +
1  1  00…… 2진코드
↓  ↓  ↓
1  0  1 …… 그레이코드
```

이와 같은 방법으로 계속하여 차례로 연산하면 그레이코드가 만들어진다. 역시 자리올림은 생략한다.

```
 +  +  +
1  1  0  0 …… 2진코드
↓  ↓  ↓  ↓
1  0  1  0 …… 그레이코드
```

또 다른 예로 2진코드를 그레이코드로 변환해 보자.

```
 +  +  +  +  +  +  +  +
1  1  0  1  0  0  1  1  0… 2진코드
↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓
1  0  1  1  1  0  1  0  1… 그레이코드
```

그레이코드로 표시된 수를 반대로 2진수로 변환하는 것도 비슷하다. 그레이코드의 MSD를 그대로 2진수의 왼쪽 첫째 자리에 놓고, 이 첫번째 자리수의 2진수와 그레이코드의 왼쪽에서 두번째 자리 비트와 더해서 두번째 2진수 자리에 둔다. 이와 같은 방법으로 한번 만든 2진수와 다음 자리수의 그레이코드와 더해 가면서 자리올림수를 생략하고 2진수를 만들어간다.

```
1  0  1  1  1  0  1  0  1… 그레이코드
↓+↗↓+↗↓+↗↓+↗↓+↗↓+↗↓+↗↓+↗↓
1  1  0  1  0  0  1  1  0… 2진코드
```

위와 같은 그레이코드를 반사 2진코드(reflected binary code)라고도 한다.


〈참고문헌〉

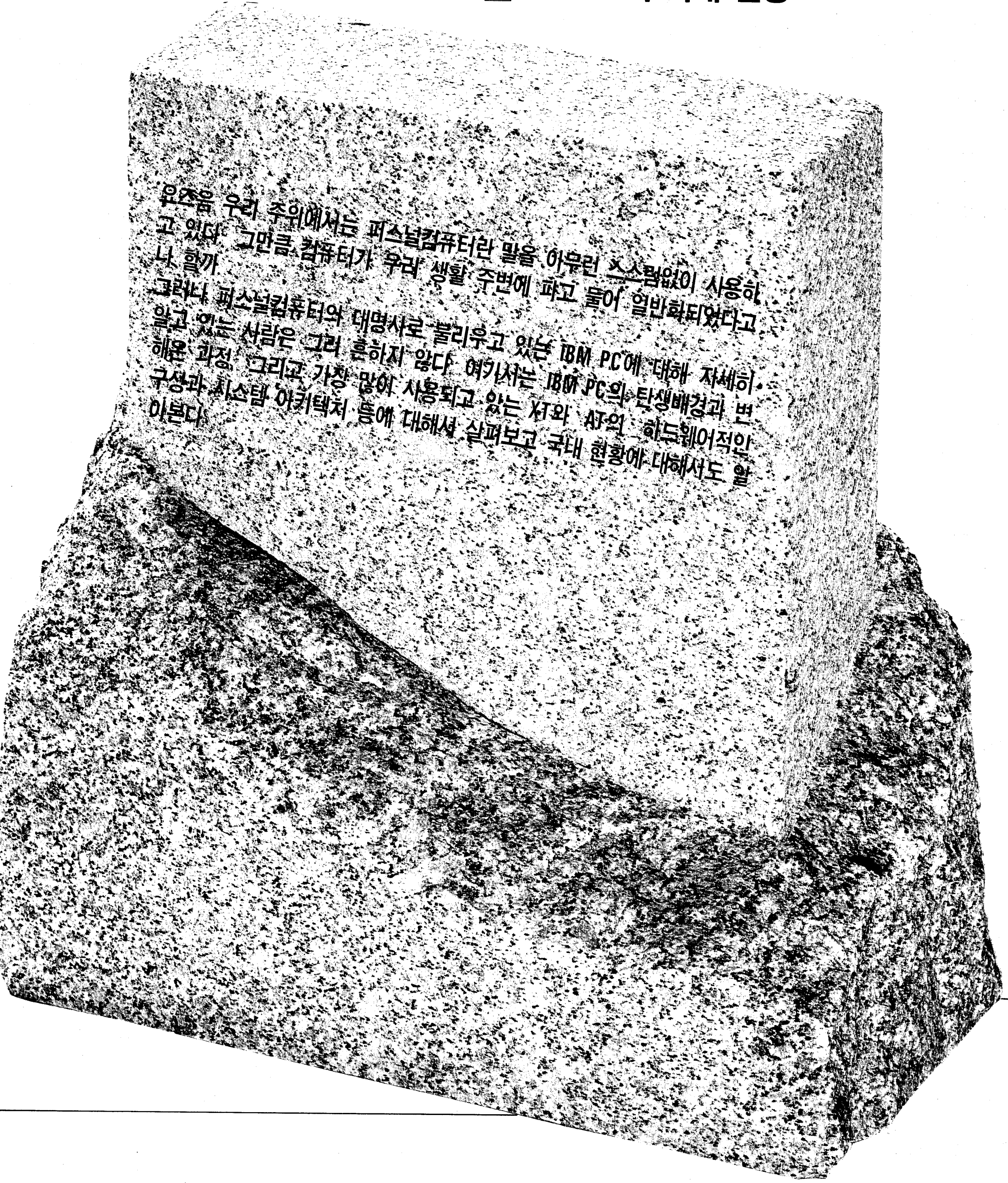
1. Charles H. Roth Jr, "Fumdamental of Logic Design", West Publishing Co., 1985.
2. M. Morris Mano, "Digital Logic and Computer Design", Printice-Hall Inc., 1979.
3. 박규태, "새 디지틀공학", 문운당, 1988.
4. W. H. Gothman, "Digital Electronies", Printice-Hall, 1977.

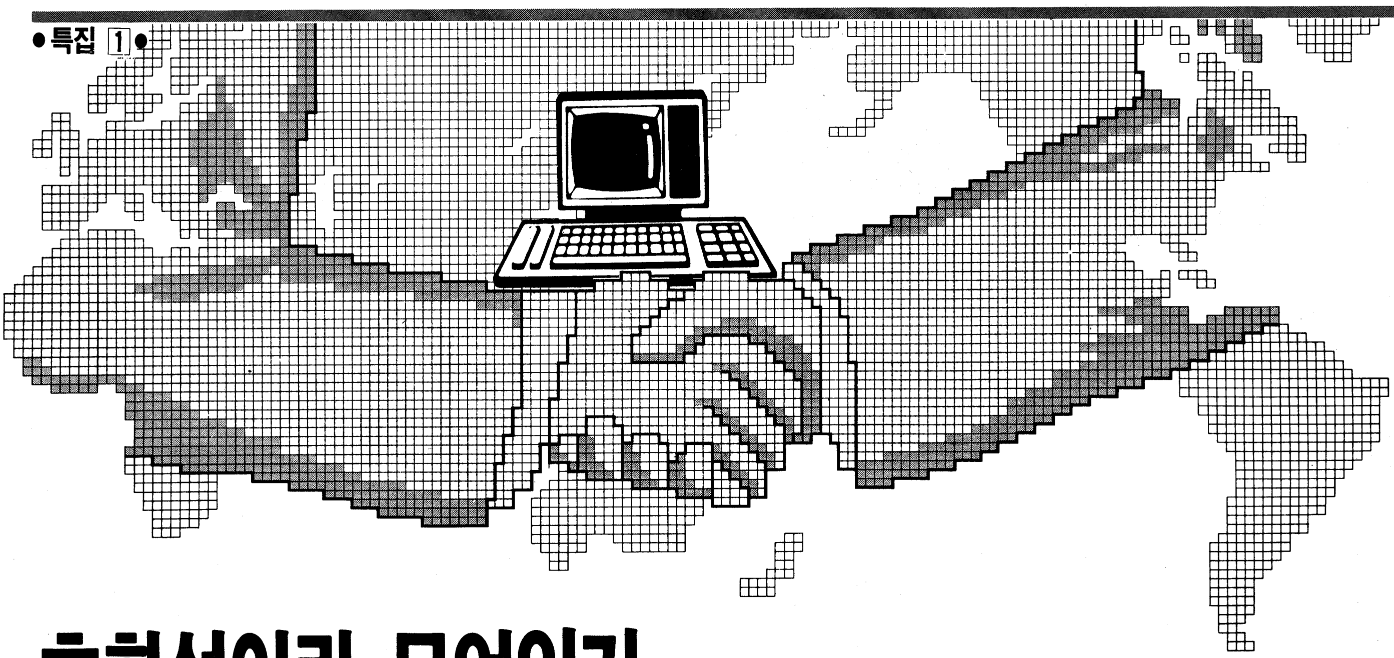
**특집**

# IBM PC 철저해부

## XT, AT 등을 비교 분석하고 국내 현황을 알아본다.

**1 호환성이란 무엇인가/ 2 XT의 하드웨어**
**3 AT의 하드웨어/ 4 IBM PC의 국내 현황**





# 호환성이란 무엇인가

IBM PC의 발표는 1981년이므로 벌써 7년이란 세월이 흘렀지만 아직까지도 세계시장에서의 IBM PC의 위치는 한마디로 말해 퍼스널컴퓨터라 하면 IBM PC를 가르킬 정도로 확고부동하다.

국제시장에 있어서도 IBM PC의 출현은 늘 역사의 한 장을 갈라놓은 것과 마찬가지로 일반인의 퍼스널컴퓨터에 대한 개념을 바꾸어놓은 계기가 되었다.

이전까지는 퍼스널컴퓨터라하면 일부의 유저들을 위한 취미적인 성격이 강해졌지만 IBM PC의 발표에 의해서 퍼스널컴퓨터가 사무용품의 취급을 받게 된 것이다. OA를 겨냥한 퍼스널컴퓨터가 그 이전에 없었는가 하면 그렇지는 않고, 예를 들면 TRS80과 같이 CP/M80을 OS로 사용하는 기종이 꽤 인기를 끈 시기도 있었지만 IBM PC의 등장에 의해서 세계시장에서 경쟁하는 다른 기종은 거의 자연스럽게 소멸하여 버렸다.

이것은 IBM PC가 기술적으로 우수하다든가 하는 의미는 아니고 IBM이라는 세계최대의 계산기 및 사무기메이커가 자사의 브랜드로 퍼스널컴퓨터를 발매하였다고 하는 것에 큰 의미가 있다고 보아야 할 것이다.

어쨌든 그 이후는 IBM PC가 사실상의 사무용 퍼스널컴퓨터의 표준으로 인식되어 이전에는 생각할 수 없었던 한기종으로의 표준화가 정착되게 되었다. 그것을 실마리로 IBM PC 를 위한 막대한 소프트웨어 개발투자가 이루어져 오늘날 IBM PC는 세계시장에서 가장많이 사용되는 기종으로 자리잡게 되었다.

IBM PC는 소프트웨어가 풍부하므로 팔린다고 하는 견해도 있지만 IBM PC의 발매와 동시에 소프트웨어가 마련되어 있었던 것은 아니다. 그럼에도 불구하고 IBM의 신용과 IBM 대형기종에서 사용되던 다수의 소프트웨어가 언젠가는 PC에도 사용되리라는 기대감으로 대부분의 사용자가 IBM PC를 선호하게 되었다.

IBM PC가 어떤 사용환경을 고려하여 설계된 것인가는 개발담당자가 아니면 알 수 없다. 그러나 초기의 기종은 **그림 1**에서 보는바와 같이 그 구성은 16비트 기종으로 어느쪽인가 하면 애플Ⅱ나 코모도로 PET에 가까운 이미지이며, 현재의 PC XT나 PC AT와는 전혀 다른 것이었다. 지금에는 이야기 거리로 되어 있지만 초기의 IBM PC에는 카세트테이프의 인터페이스 코넥터가 실장되어 있었고 16K바이트 + 롬베이직이라는 최소 구성으로 현실적인 응용에는 너무나도 미약한 것이었다.

이것은 명백하게 호비스트, 단적으로 말하여 애플유저층을 겨냥하여 만들지 않았나 생각된다. 그 후에 발표한 유명한 실패작 PCjr의 규격에도 IBM이 그러한 개념을 지향하고 있었던 것을 느끼게 한다. 그런데 당초의 예상에 반하여 OA쪽 시장이 확대되고 있었다고 하는 것이 실제의 실정이었다. 따라서 초기의 IBM PC 제품들은 생각밖으로 많은 호응을 받았다. 이 초기 IBM PC의 영업적인 성공에 힘입어 PC XT, PC AT 그리고 PC/2로 확장되어 그것이 그대로 **그림 2**와 같은 IBM PC의 역사로 되어 있는 것은 누구나가 잘

알고 있는 것이다.

이 초기 PC의 규격은 표 1에 나타낸 바와 같이 그 후의 PC패밀리에도 계승되어 있어서(그것이 세계최대의 계산기메이커의 신용이라고도 할 수 있지만) 대부분의 소프트웨어, 하드웨어는 PC패밀리이면 어느 것으로도 사용할 수 있도록 되어 있다.

그러나 호환성을 유지하기 위해서 기종간의 특성을 무시한채 희생되는 부분도 많았다. 따라서 PC AT는 그것 이후의 PC/2와는 호환성을 거의 가지지 못하고 있다. 그러한 의미에서는 IBM PC AT라는 것은 IBM PC가 다다른 최후의 모습이라 할 수 있을 것이다.

# IBM PC의 하드웨어

## • PC XT의 하드웨어

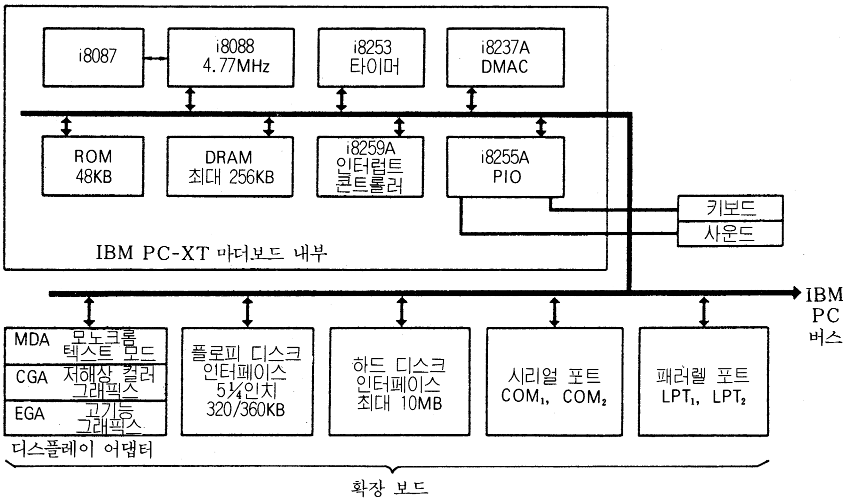
IBM PC는 한마디로 말해서 극히 조심스러우며 소극적으로 만들어진 MS-DOS머신이다. 그렇다고는 하지만 1981년에 발표된 당시의 표준 규격은 본체내 마더보드의 메모리가 64K바이트이며 하드디스크의 서포투도 없어 본격적인 사무처리에는 불충분하였다.

IBM은 1983년에 그 때까지의 IBM PC의 규격을 재검토하여 PC를 PCjr과 PC XT의 기종으로 나누었다. PC XT는 그때까지 사용하고 있었던 16K DRAM을 64K DRAM으로 변경하여 표준에서 320K바이트 플로피디스크를 탑재하였으며 새로이 개발된 PC DOS2.0을 지원하였다. 이것은 사무처리분야를 겨냥한 포석이었다.

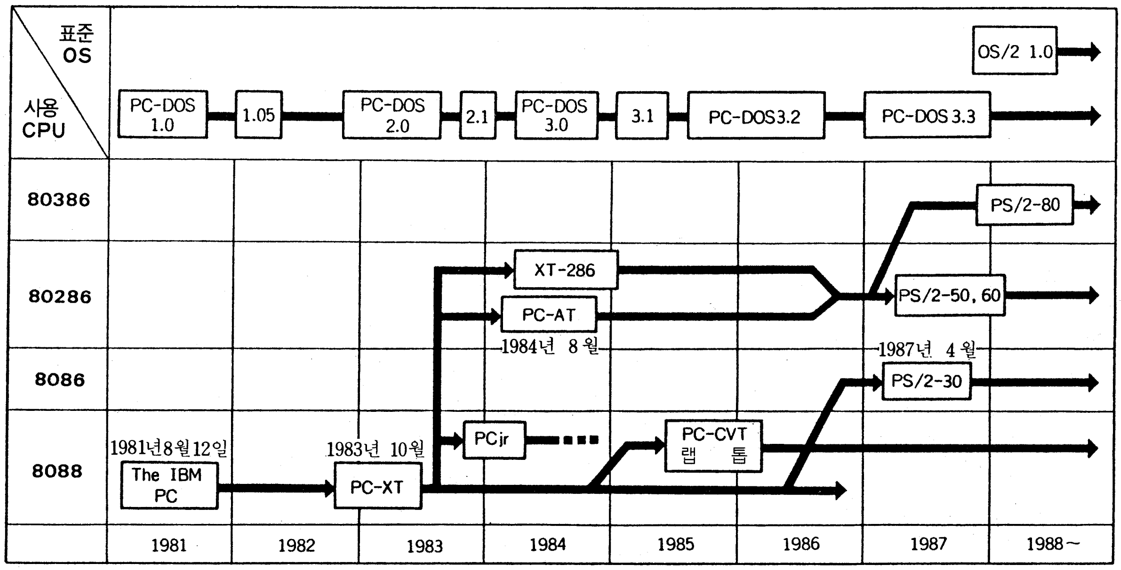
이것에 비해서 PCjr은 PC의 확장성을 대폭으로 제한하였다. 메모리는 최대 256K바이트까지, 키보드는 값이 싼 고무패드를 사용한 것으로 된 반면 조이스틱이나 그래픽스보드는 표준에서 부속시킨다고 하는 홈컴퓨터의 성격이 강한 것이다.

유감스럽지만 PCjr은 너무 빈약하였기 때문에 유저가 받아 들이지 않아서 이후는 PC XT가 IBM PC의 표준기종으로 되었다. 단지 IBM PC라고 하는 경우는 초기의 PC 및 PC XT를 가리키지만 이 구성은 실로 단순하며, 마더보드에 실려있는 LSI

〔그림 1〕 IBM PC-XT 규격




〔그림 2〕 IBM PC 패밀리의 변천도



〔표 1〕 IBM PC 주요 기종의 기본규격

| | IBM PC | PC XT | PC AT |
|---|---|---|---|
| CPU | 8088 | 8088 | 80286 |
| Clock(MHz) | 4.77 | 4.77 | 6/8 |
| 최대 주기억 용량 | 640kB | 640kB | 15MB |
| 내부 버스폭 | 8 | 8 | 16 |
| 외부 인터페이스 버스폭 | 8 PC-Bus | 8 PC-Bus | 8/16 PC-Bus(8), AT-Bus(16) |
| 조립 플로피 디스크 | 5.25인치 | 5.25인치 | 5.25인치 |
| 표준 플로피 용량 | 320kB | 320kB | 1.2MB |
| 하드 디스크 용량 | — | 10MB | 20/30MB |
| 표준 OS | PC-DOS1.0 | PC-DOS2.1 | PC-DOS3.1(OS/2 가능) |
| 발표년도 | 1981 | 1983 | 1984 |

는 CPU와 메모리를 제외하면 8253, 8255A, 8259A, 8237A가 각 1개인 단순한 것이었다.

IBM PC가 다른 퍼스널컴퓨터와 결정적으로 다른 것은 디스플레이 인터페이스가 옵션이라는 점이었다. 유저는 구입시에 목적에 따라서 표 2의 MDA, CGA, EGA 가운데서 선택할 수가 있었다. 이것은 IBM PC의 사용형태에 고정개념을 부어 넣지 않는다고 하는 의미에서 결과적으로 큰 성공이었다고 생각한다.

## • PC XT의 하드웨어

1984년에 IBM은 처리속도가 향상한 IBM PC를 PC AT로 발표하였다. PC AT가 그때까지의 PC와 다른 점은 XT 까지의 IBM PC에서는 8비트 외부버스의 8088CPU를 채용하고 있었던 것을 PC AT에서는 80286을 채용하여 16비트 내부버스로 이행하여 80286의 프로텍트 모드에 의해서 16M바이트까지의 기억공간을 서포트가능하게 한 점에 있다.

XT까지의 IBM PC는 4.77MHZ의 8비트 버스의 8088을 사용하고 있었으므로 6MHZ의 80286과 16비트 버스에 의한 CPU처리속도의 상승은 단순한 벤치마크에서는 2.5 배 정도의 성능향상을 가져온다. 주변기기의 서포트의 면에서도 플로피디스크 드라이브가 320/360K바이트에서 1.2M바이트로 하드디스크도 XT의 10M바이트에서 20/30M바이트로 강화되어 종합적인 처리능력은 비약적으로 향상하였다.

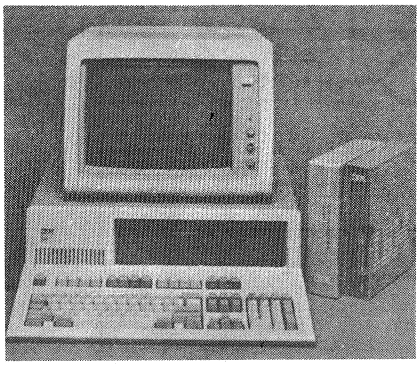
외부버스사양의 면에서는 PC버스라고 불리우는 8비트 버스를 계승한 위에 80286의 16비트 버스사양을 AT버스로서 추가하여 고속의 주변기기 지원이 가능하게 하고 있다. 이들의 사양 강화에도 불구하고 PC AT와 PC XT 와의 호환성은 극히 높고, 주변LSI도 16비트 버스를 위해서 칩수를 증가시켰을 뿐 종류는 XT의 그것과 같다. 그 때문에 XT용의 소프트웨어, 하드웨어가 거의 변경없이 사용될 수 있었다.

시스템개발의 입장에서 말하면 PC AT의 최대의 이점은 메모리공간의 확장에 있는 것이지만 AT의 표준OS인 PC DOS3.X는 PC DOS2.X와 같은 640K바이트까지의 메모리공간밖에 지원되지 않고, 사실상

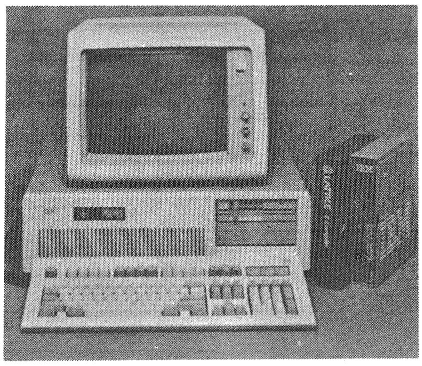
〔표 2〕 IBM PC의 디스플레이 규격

| 형　식 | 화면구성 | 최대표시색 | 비　고 |
|---|---|---|---|
| MDA (모노크롬 디스플레이 어댑터) | 80×25문자 | 1 | 어트리뷰트 8비트 |
| CGA (컬러 그래픽스 어댑터) | 80×25문자<br>40×25문자<br>640×200 화소<br>320×200<br>160×100 | 16<br>16<br>1<br>4<br>16 | MC6845CRTC<br>버퍼 16kB |
| EGA (엔헨스트 그래픽스 어댑터) | 80×25문자<br>40×25문자<br>640×350화소<br>640×200<br>320×350<br>300×200 | 16<br>16<br>16<br>16<br>16<br>16 | 커스텀 IC,<br>버퍼 256kB<br>CGA 에뮬레이션 가능 |

〔사진 1〕 PC XT의 외관



〔사진 2〕 PC AT의 외관



640K바이트의 벽을 넘는 메모리영역을 사용하는 응용프로램의 개발이 불가능하게 되어 있다. 또 80286의 구조적인 결함이라 할 수 있는 것이지만 일단 프로텍트모드로 이행하면 하드웨어 리세트 이외에 리얼모드에 복귀하는 수단이 없기 때문에 MS--DOS 의 관리하에서는 프로텍트 모드를 사용하지 못한다.

PC AT의 확장 메모리공간을 이용하기 위해서는 프로텍트 모드를 지원하는 OS가 불가결인 것이지만, 그것은 IBM PC에서 사용되는 수많은 소프트웨어들을 포기해야 하는 것을 의미하므로 그렇게 간단하게 생각할 수 없다. 현재 80285프로텍트 모드를 지원하는 OS로서는 마이크로소프트사의 제닉스와 OS/2가 발표되고 있지만 OS/2는 PC DOS와의 호환을 보증한 위에 프로텍트 모드의 사용이 가능하게 되어 있기 때문에 OS/2가 앞으로 PC AT 및 PS/2이후의 하드웨어에서 표준적으로 사용되리라 예상된다.

그러나 PC AT를 PC-DOS3.× 의 아래에서 사용하는 한에 있어서는 그 모델은 고속의 PC XT 이외의 아무 것도 아니라고 말할 수 있다.

## • 확장용 보드를 위한 입출력버스규격

IBM PC는 XT까지 PC버스로서 62핀의 코넥터에 의해 데이터 8비트, 어드레스 20비트의 TTL레벨의 입출력버스를 사용하고 있다. 이것은 8088(4.77MHZ)의 최대크기모드의 버스사양 그것이므로 인텔계의 I/O 칩을 접속하는 것은 쉽다. PC AT는 이 PC버스규격의 8비트 버스를 계승한 위에 AT버스로서 더욱 16비트 데이터버스, 24비트 어드레스 버스를 지원하는 사양으로 확장되었다.

AT버스는 PC버스에 확장코넥터로서 36핀의 코넥터를 추가한 것이외 기본 8비트전송과 16비트전송의 전환을 이 확장코넥터의 MEMCS16, -IOCS16을 액티브로 함으로써 가능하게 하였다. 또 IBM PC는 전통적으로 메모리, I/O사이클에 웨이트 사이클을 넣고 있었지만 AT버스에서는 이 웨이트를 제거하는 고속모드(웨이트 모드)가 -OWS입력에 의해서 가능하게 되었다.

AT버스에서는 더욱 내부CPU를 완전히 끊고 버스의 제어를 제거하는 기능(Master)이 설정되었다. 이것에 의해서 CPU를 80386등으로 전환하는 것도 확장보드측에서 가능하게 된다. PC버스와의 호환은 62핀 코넥터의 경우 타이밍(6/8MHZ 80286)을 제외하고는 완전히 동일하다. 즉 AT버스에 추가된 제어신호는 입력에 대해서는 내부에서 풀업되어 있어서 사용하지 않으면 PC 버스모드로 동작하는 것이 보증되어 있는 것이다.

또 IBM은 현재 PC AT의 제조를 중지하고 AT급의 기종을 PC/2모델50, 60 으로 교대시키고 있다. 아까운 것은 PS/2모델50이상에서는 AT버스를 지원하고 있지 않아서 새삼스럽게 마이크로채널 버스사양이 정의되었다. 이것은 80386을 의식한 고속전송이 가능한 것이며 그 의미에서도 AT버스사양이 앞으로 사용되지 않게 될 것으로 예상된다.

## • IBM PC패밀리의 I/O맵

80×86CPU는 메모리공간과는 별도로 16비트의 I/O공간을 지원하고 있으며 IBM PC도 이 I/O공간에 주변기기를 배치하고 있다. 단, 주의하지 않으면 안되는 것은 IBM PC에서는 마더보드내의 I/O칩 디코드시에 $A_{15}$~$A_{10}$의 6비트가 무시되고 있는 점이다. 따라서 유효한 I/O포트수는 1024(0~3FFH)로 된다. 이 중 0~1FFH는 마더보드내에 예약되고 있다. 그 위에 어드레스 공간에 대해서도 거의 확장보드에 예약되어 있어서 유저가 자유로 사용할 수 있는 공간으로서 해방되어 있는 어드레스는 0250H~0277H, 0280H ~2EFH, 0330H~0377H, 0390~039FH로 되어 있다. 또 메모리공간에 대해서는 0~A000H의 640K바이트에 대해서는 MS-DOS가 점유하고 있으며, A000H 이상의 384K 바이트에 비디오 인터페이스, ROM--BIOS, 디바이스 드라이버 등이 배치된다. PC AT에서는 더욱 24비트 어드레스공간이 지원되는 것이지만 MS-DOS에서는 액세스할 수 없기 때문에 현실적으로 사용할 수 없다.

실제의 구성에서 비디오 인터페이스는 MDA, CGA, EGA의 어느 것인가가 실장되므로 메모리공간의 어딘가에 빈 곳이 생길 것이지만 최근에는 빈자리가 확장메모리(LIM-EMS:컬럼 참조)를 위한 윈도우로서 사용되므로 빈 곳은 없다고 생각한 쪽이 무난하다.

## • 소프트웨어 인터페이스

IBM PC상의 응용 프로그램은 하드웨어자원을 액세스하는 경우에 원칙적으로 MS-DOS의 시스템코일(INT20H~INT3FH) 및 ROM-BIOS코일(INT0H~INTFFH)을 경유하지 않으면 안된다. MS-DOS의 시스템코일은 하드웨어의존성이 없으므로 이 레벨에서 인터페이스된 경우의 호환은 극히 높아진다.

그러나 화면제어나 세밀한 하드웨어의 제어를 할 수는 없다. 실제로 유통되고 있는 소프트웨어는 대부분 ROM-BIOS를 사용하고 있다. 따라서 IBM PC용의 소프트웨어를구입하여 그것을 MS-DOS를 사용하는 호환기종에서 사용할 경우 100% 전부 동작되지는 않는다.

또 하나의 중요한 인터페이스규격

은 플로피디스크의 형식이다. 현재 IBM PC가 지원하는 플로피디스크의 포맷은 **표 3**과 같다. 양면배밀도(1.2M 바이트)형식은 AT의 표준포맷이며 IBM PC패밀리간에서는 양면배밀도(8섹터 포맷, 320K바이트) 형식이 가장 표준적으로 사용되고 있다.

이른바 PC호환기종 IBM포맷이외에 여러가지의 포맷을 지원하는 것이 많지만 이들 가운데에는 IBM PC에서는 지원되지 않은 것이 있으므로 소프트웨어의 이식 등에서는 충분히 주의하는 것이 좋으며 일반적으로는 320K포맷이 안전하다.

**〔표 3〕 IBM PC가 지원하는 플로피 디스크의 기록 포맷**

| PC/DOS 버전 | 디스켓의 버전 | 총트랙수 | 섹터/트랙 | 바이트/섹터 | 용 량 |
|---|---|---|---|---|---|
| PC-DOS1.1 | 편면배밀 | 40 | 8 | 512 | 160KB |
| | 양면배밀 | 80 | 8 | 512 | 320KB |
| PC-DOS2.0, 2.1 | 편면배밀 | 40 | 8 | 512 | 160KB |
| | | 40 | 9 | 512 | 180KB |
| | 양면배밀 | 80 | 8 | 512 | 320KB |
| | | 80 | 9 | 512 | 360KB |
| PC-DOS3.0, 3.1, 3.2 | 편면배밀 | 40 | 8 | 512 | 160KB |
| | | 40 | 9 | 512 | 180KB |
| | 양면배밀 | 80 | 8 | 512 | 320KB |
| | | 80 | 9 | 512 | 360KB |
| | 양면배밀배트랙 | 160 | 9 | 512 | 720KB*1 |
| | 양면고밀도 | 160 | 15 | 512 | 1.2MB*2 |

*1 720KB 포맷은 PC 콤패티블용 3.5인치 디스크의 포맷으로 5인치에는 정식으로는 사용되지 않는다.
*2 PC XT의 표준규격

●컬럼A●

## LIM-EMS

지금 IBM PC가 안고 있는 문제의 하나는 유저가 사용할 수 있는 메모리공간이 640K바이트라고 하는 점이다. 8086의 어드레스공간은 1M바이트이지만 A000H의 상위 영역에는 비디오인터페이스나 상주ROM이 예약되어 있기 때문에 MS-DOS는 지원하지 않는다. 옛날은 그것만으로도 충분한 용량이라는 인식때문이었겠지만 그후 DRAM의 발달과 응용 프로그램의 대용량화에 수반하여 640K바이트를 넘는 기억영역의 지원 요구가 강하게 되어 왔다.

PC AT는 하드웨어적으로는 XT에 비해서 많이 개선되었지만 가장 중요한 OS는 바뀌지 않았다. 국산 퍼스널컴퓨터도 MS-DOS를 채용한 시점에서는 마찬가지의 문제를 안고 있다. 상식적인 대응책은 뱅크메모리 개념을 도입하는 것이다.

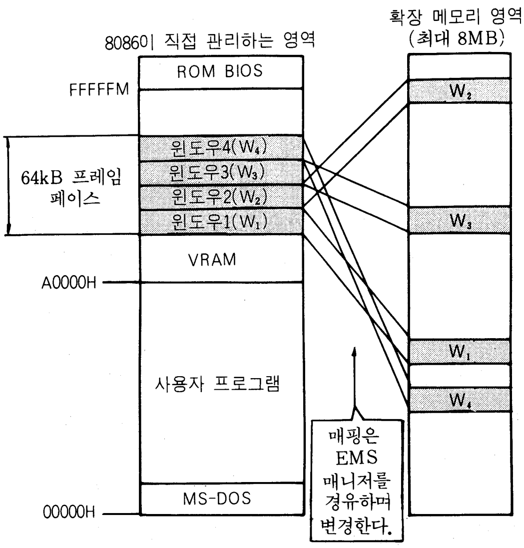
IBM PC에서도 마찬가지의 수법으로 640K바이트를 넘는 메모리를 액세스하지만 액세스의 방법과 지원 소프트웨어는 공식으로 규격화되어 있다. LIM-EMS는 Lotus-Intel-Microsoft Expanded Memory Specification의 약자로서 인터페이스시 하드웨어에 의존하지 않고 640K바이트를 넘는 기억영역을 사용할 수 있다고 하는 구조이다.

LIM-EMS에서는 페이지프레임이라고 불리우는 64K바이트의 연속영역(보통은 C000H 보다 상위영역)을 사용하여 뱅크 메모리를 액세스한다(그림A). 페이지프레임은 더욱 4개의 16K바이트의 윈도우로 분할되어 각각 독립으로 확장메모리내에 매칭된다. IBM PC에서는 A000H 이상의 공간은 전부 시스템에 예약되어 있으므로 어디에 페이지프레임을 설정하는가는 그 때의 하드웨어 구성에 의존한다.

그래서 이 뱅크메모리를 논리적인 장치에 추상화하는 것이 LIM-EMS매니저라고 불리우는 확장시스템 코일(INT 67H)이다. LIM-EMS매니저를 경유해서 확장메모리를 액세스함으로써 최대 8M바이트 공간을 MS-DOS에서 사용할 수 있다.

IBM PC에서의 램디스크는 거의 이 인터페이스를 경우하여(1개의 윈도우를 사용한다) 이루어지고 있다. 페이지프레임은 4개의 윈도우로 되므로 램디스크를 사용한 경우에는 나머지의 윈도우는 응용프로그램을 사용할 수 있다.

〔그림 A〕 LIM-EMS 규격의 확장 메모리 인터페이스



## IBM PC의 OS

그림 2 에도 나타낸 바와 같이 IBM PC의 역사는 그대로 PC/MS-DOS의 역사라 할 수 있다. PC-DOS와 MS-DOS는 구조적으로는 동일하다고 보아도 되는 것이지만 마이크로소프트사가 IBM과의 개발계약에 의해서 IBM PC전용으로서 작성한 것을 PC-DOS라 부르고 있다. 마이크로소프트는 개발계약에 의해서 작성된 소프트웨어를 IBM과는 독립으로 판매하는 권리를 보존하고 있기때문에 PC-DOS와 동일한 기능의 OS가 IBM이외의 하드웨어용에 MS-DOS로서 공급되고 있는 것이다.

MS-DOS는 기능적으로는 싱글유저, 싱글태스크의 OS로 분류된다. 확실이 MS DOS는 화일관리, 입출력자원관리, 기억관리, 프로그램의 기동, 종료라고 하는 OS에 요구되는 최저한의 기능을 지원하지만 그 레벨은 유닉스 등의 그것과는 비교도 되지 않을 정도로 빈약하다.

이것은 8088/86이라는 저레벨의 CPU와 IBM PC라는 고정된 하드웨어의 시스템에 구속되어 작성되어 있는 것도 원인으로 들 수 있다.

그러나 저레벨이기 때문에 유저프로그램에서 IBM PC의 하드웨어 자원이나 OS의 일부까지도 직접 언급할 수 있어서 결과적으로 IBM PC의 빈약한 하드웨어에서도 고기능인 소프트웨어를 사용할 수 있게 된 것이다. 그러한 의미에서는 IBM PC의 수준에서는 MS-DOS는 알맞는 OS였다고 할 수 있다.

그러나 PC AT가 등장한 시점에서 보면 MS-DOS의 기능적인 힘의 약함이 도움이 되어 온것은 사실이다. MS-DOS는 네트워크기능(3.1판)이나 윈도우시스템(MS-Windows) 등으로 보완하였지만 근본적인 해결책은 되어 있지 않는 것 같다. MS-DOS에서 PC AT이상의 하드웨어자원을 유효하게 이용할 수 있는 OS의 필요성은 이전부터 요구되어 왔고, 마이크로소프트사에서도 멀티유저, 멀티태스크의 OS로서 제닉스(유닉스)를 정식으로 PC AT에 지원하고 있지만 종래의 MS-DOS상 소프트웨어와의 호환성이나 유닉스를 사용하는데에는 하드웨어의 빈약함으로 인해 그 보급은 그다지 진행되고있지 않다.

IBM은 1987년 후반에 IBM PC의 신시리즈로서 Personal System/2를 발표하였지만 거기서 요구되는 OS는 종래의 MS-DOS베이스의 소프트웨어자산을 계승하는 것이 가능하고 또한 80286/386의 기능을 충분히 활용도록 되어 있다.

즉 MS-DOS와의 호환을 유지하고 IM바이트이상의 기억공간을 이용할 수 있고, 더욱 멀티태스크, 멀티유저환경을 제공하는 OS이다. 즉, MS-DOS3.2, OS/2 그리고 유닉스를 비교하면 표 4와 같이 된다. 이것은 보는 견지에 따라서는 MS-DOS라고 하기 보다는 유닉스에 가깝고 혹은 그것을 넘는 것도 있다.

확실히 MS-DOS도 2.0판 이후에는 유닉스의 기능을 첨가하고 있지만 어디까지나 표면적으로는 OS/2는 MS-DOS와는 전혀 다른 것으로 생각하는 쪽이 좋을 것이다.

## IBM PC의 소프트웨어

IBM PC는 결과적으로 많은 호환메이커와 많은 소프트웨어 라이브러리를 갖고 있다. 확실히 말하면 대부분의 PC용 프로그램이 IBM PC상에서 움직이고 있는 것이다. 그 가운데에서도 IBM PC와 가장 밀접한 것이 로터스의 1-2-3이다.

로터스 1-2-3은 기본적으로는 비지캘크의 흐름을 추종하는 스프레드쉬트인 것이지만 IBM PC의 흐름에 실린 것과 통합화소프트웨어라고 하는 개념의 도입으로 지금까지 주류인 비지캘크나 수퍼캘크라고 하는 애플 CP/M계의 스프레드 쉬트시장을 단숨에 IBM을 중심으로 바꾸어 버렸다.

1-2-3은 IBM PC가 OA시장에서 기대되고 있었던 사용방법, 즉, 데이터입력, 집계, 분석 그리고 도큐먼트화라는 일련의 처리를 하나의 프로그램으로, 통일된 환경에서 행하는 것을 처음으로 가능하게 한 것

〔표 4〕 MS-DOS, OS/2, 유닉스의 비교

| | MS-DOS3.2 | OS/2 | Unix |
|---|---|---|---|
| 멀티 태스크 | 불가능 | 가능 | 가능 |
| 멀티 유저 | 불가능 | 불가능 | 가능 |
| 프로세스간 통신 | 없음 | pipe, 공유 메모리, queue | pipe, 공유 메모리, 세머포, queue |
| 가상기억 | 없음 | 있음 | 있음 |
| 전원관리 | 없음 | 화일록 가능 | 화일록 가능 |
| 화일 시스템 | 계층 디렉토리 | MS-DOS3.×와 동일 | 계층 디렉토리, 네트워크 화일 |
| 유저 인터페이스 | MS-Windows | 프리젠테이션 매니저 | Xwindows, NeWs etc. |
| 네트워크 | MS-Net | MS-Net | 이더네트 |
| 기타 | | 다이나믹 링킹, MS-DOS호환성 | |

이다.

또 하나의 큰 공헌은 MASM, MS-FORTRAN 으로 대표되는 마이크로소프트의 본격적인 개발툴이다. PC-DOS에서는 당초 응용프로그램은 그렇게 많지는 않았으나 대규모 프로그램의 개발에 필수인 툴이 갖추어져서 소프트웨어 개발환경이 CP/M86과 비교하여 훨씬 풍부하였다.

IBM PC가 발매된 당시는 DOS도 옵션이였으므로 유저는 자기의 기호에 맞는 CP/M86과 PC-DOS를 선택하였다.

많은 유저가 PC-DOS로 흐른 사정에는 PC-DOS가 값이 싼 것도 이유지만 당시 MS-DOS에서는 8087을 지원하는 풀세트의 포트란77컴파일러가 지원된다고 하는 사실이 기술계의 유저들에게는 큰 매력으로 작용하였기 때문이다.

IBM PC의 사양은 OA용으로는 충분하여도 기술계 소프트웨어로는 다소어딘가 부족하다는 것은 누구나가 인정하는 것이다. 특히 그래픽스의 사양이 최상위기종에서도 640×350(EGA)인 것과 PC-DOS의 유저공간이 640K바이트인 것은 CAD나 DTP 에서는 불충분하다고 이야기되고 있다.

또 현재의 PC에서는 본격적인 그래픽스인터페이스 특히 윈도우시스템과 같은 고레벨의 표준인터페이스가 충분하지 못하다. MS-Windows가 표준인터페이스로서 제공되기는 하나 주기억용량이 640K바이트라는 제한으로 인해 작업영역의 제한을 받으며 그것을 사용한 본격적인 응용 소프트웨어는 거의 눈에 띄지 않는다. 그럼에도 불구하고 AUTOCAD가 PC상의 CAD 표준으로 사용되고 있다. 이들의 분야에는 PC/2의 상위기종이 본격적으로 보급되리라 생각된다.

## PC 호환기종

IBM PC가 퍼스널컴퓨터의 표준기로된 시점에서 퍼스널컴퓨터의 목표는 실질적으로 크게 2개로 나누어졌다. 즉 IBM PC시장과 비호환시장이다. 비호환시장쪽은 애플을 필두로 코모도로 아타리 등의 기종이다. 최근에 와서 매킨토시가 상위기종의 PC로서 DTP시장에 침투하고 있지만 OA용의 퍼스널컴퓨터는 IBM PC가 독점하다시피 되어 버렸다.

당연하지만 신규로 퍼스컴시장에 참가하는 경우는 IBM PC가 표준기로 정착된 이상 이것과 전혀 다른 것은 유저가 받아 들이지 않으므로 자연히 IBM PC와 유사한 것으로 되지 않을 수 없다. 또 IBM PC는 하드웨어, 소프트웨어의 도큐먼트가 극히 상세하게 공개되어 있어서, 동일사양의 것을 만드는 것이 쉬운 배경도 가지고 있어 대부분의 회사가 호환기종을 생산하고 있다. 보통 IBM 제품은 각 파트에 자사규격품을 사용하기 때문에 호환제품을 만드는 경우에 부품레벨에서 만들지 않으면 안되는 것이 많지만 IBM PC는 예외적으로 범용부품이 나오고 있기 때문에 점점 만들기가 쉽다.

한마디로 PC호환이라고 하여도 전부 동일하면 저작권 등의 문제가 생기므로 어딘가가 다르게 되어 있다. 특히 IBM PC는 사양적으로 그 시점의 최고수준은 아니고 극히 무난한 설계이므로 성능적으로 상회하는 것도 용이하여 설계상의 미비한 점도 호환기종에서는 고쳐서 제조할 수 있다.

그러나 그것도 한마디로 단정할 수 없으며 IBM PC의 버그에서 우연히 움직이고 있는 하드웨어나 소프트웨어가 있었던 경우에 그러한 개량은 호환성을 내리는 것으로 된다. 즉,"버그도 사양"이라는 이야기가 되는 경우이다. 간단히 말하면 90%의 호환성에서는 50%의 소프트웨어밖에 움직이게 하지 않는다고 생각하는 쪽이 좋을 것이다. 90%의 소프트웨어를 움직이게 하려면 99.9%같지 않으면 안된다. 이 점에 PC 호환기종의 어려움이 있다.

PC호환기종의 유사도는 크게 4단계로 분류할 수 있다.

**① PC**

조잡한 것, 단순한 MS-DOS를 사용하는 것, IBM PC의 디스크는 읽을 수 있도록 되어 있는 것이 많지만 표준포맷은 다른 것이 많다. CPU도 달라졌다든지 하는 습관으로 외관만 IBM PC와 닮아 있는 것이 있으므로 요주의.

**② PC 데이터레벨 호환**

IBM PC와 플로피 레벨에서 데이터교환을 할 수 있는 것, 하드웨어의 규격은 다르기 때문에 확장기판은 IBM PC용의 것을 사용할 수 없는 것이 많다. 상당한 수의 소프트웨어 사용은 가능하지만 그 기종용으로 만들어진 소프트웨어를 사용하는 것이 무난.

**③ PC 기능레벨 호환**

IBM PC버스를 가지며 CPU도 같지만 보통은 클럭을 올리고 있다든지 버스폭이 내부16비트였다든지 한다. IBM PC용의 확장기판은 그대로 사용할 수 있고, 소프트웨어도 IBM용의 것을 그대로 사용할 수 있는 것이 많다. 외관은 독자성을 주장하여 일련 IBM PC로 보이지 않는 것이 많다. ROM-BIOS의 호환성을 유지하고 있다.

**④ PC 조작레벨 호환(PC클론)**

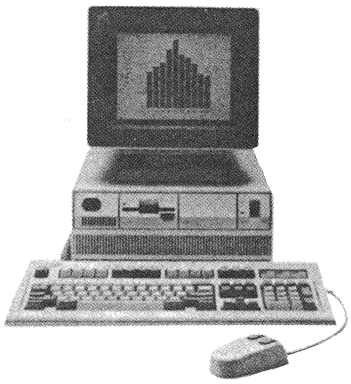
CPU의 클럭까지 IBM PC와 같은 IBM이 제공하는 소프트웨어(OS를 포함), 하드웨어까지 개조없이 움직인다. BIOS-ROM은 표준에서 Phoenix BIOS가 붙어 있지만, IBM 순정ROM-BIOS로 바꾸어도 움직인다.

이상의 분류는 일반적인 분류이며

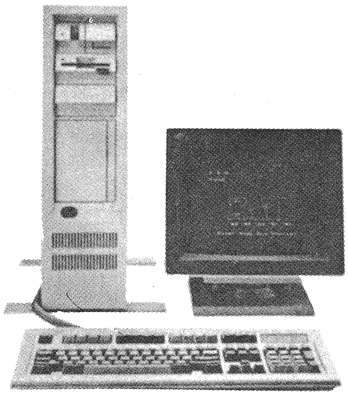
〔사진 3〕 PS/2 MODEL30



〔사진 4〕 PS/2 MODEL 50



〔사진 5〕 PS/2 MODEL 80



명확한 정의가 있는 것은 아니다. 이 중에서 이른바 PC클론이 가장 호환성이 높다.

PC 기능레벨 호환의 일군은 주로 소프트웨어의 면에서 IBM PC와 호환성이 높고 성능면에서 개선하기 위해서 클럭을 올리기도 하고, 디스크를 고밀도로 한다든지 하는 것이 많다.

또 일련 호환이라 하지만 거의 호환성이 없는 기종도 있다.

IBM PC는 기본적으로는 80×86/88CPU에 의한 MS-DOS머신이라고 볼 수 있으므로 CPU로 80×86/88을 채택한 기종은 호환기종이라고 말할 수 있다.

그러나 이 레벨의 호환성에서 IBM PC의 소프트웨어는 전혀 움직이지 않는다고 하여도 과언은 아니다. 문제는 MS-DOS의 시스템코일이외의 시스템코일, 즉 ROM-BIOS 코일에 대응할 수 있는지 어떤지이다. PC호환을 칭하기 위해서는 최소한 ROM-BIOS레벨의 인터페이스가 없어서는 안된다.

호환성에 대해서도 문제가 되는 것은 저작권, 특히 소프트웨어저작권이다. MS-DOS에 관해서는 정식으로 입수할 수 있지만 IBM순정의 ROM-BIOS는 사용할 수 없다. 그러나 그 사양 및 소스코드는 기술자료로서 공개되어 있다. 그래서 일반적으로 호환기에서는 기능적으로 동일하지만 전부 독립으로 개발된 ROM-BIOS 를 탑재하고 있다. 가장 저명한 ROM-BIOS는 Phoenix BIOS라고 불리우는 것이며, 소프트웨어 인터럽트에 의한 인터페이스에 관해서 거의 완전하게 호환이라고

●컬럼B●

## OS/2와 MS-DOS의 호환성

OS/2의 당면의 문제는 80286의 프로텍트모드를 지원하면서 MS-DOS의 응용프로그램이 동작하는 점에 있다.그러나 MS-DOS의 응용프로그램의 동작은 OS/2의 멀티태스크환경하에서 움직이게 하는것은 아니다. MS-DOS의 응용프로그램에 관해서는 어디까지나 1개의 프로세스밖에 진행이 되지 않으며 동시에 움직이지 않는다.

OS/2의 MS-DOS호환모드는 호환박스, 혹은 3.X박스라고 불리우지만 이 모드에서는 80286이 8086모드에서 사용되도록 되어 있다. 즉 MS-DOS 호환 모드에서의 PS/2의 메모리공간 최저부 1M바이트는 종래의 PC와 같은 할당으로 되어 있어서 리얼모드로 스위치하면 PC XT 등과 같다고 볼 수 있는 것이다.

OS/2는 이 부분에 MS-DOS의 에뮬레이터를 배치하고 있어서 프로텍트모드의 응용프로그램은 이 영역은 사용할 수 없다. 따라서 프로텍트 모드의 프로그램은 전부 1M바이트보다도 위로 움직인다. 이것이 OS/2에서는 최저에서도 2M바이트가 요구되는 이유이다.

그런데 기술적으로 흥미가 있는 것은 OS/2가 어떻게 해서 프로텍트모드와 리얼모드를 할 수 있는 것인가 하는 것이지만 이 점에 대해서는 아직 알려지지 않고 있다.

이것이 알려지지 않고 있는 이유는 OS/2의 개발원인 마이크로소프트사가 이 부분에서 IBM 과는 독립으로 특허를 5개 신청하고 있어서 그 권리의 확정까지는 극비로 되어 있는 것이 주 원인이다. 또 그 때문에 OS/2의 공개가 1987년말까지 늦어진 것도 잘 알려져 있다. 일반적으로 소프트웨어에 관한 특허는 취득이 곤란하다고 이야기되고 있으므로 어떤 내용의 특허인가에 흥미가 모아지고 있다.

이야기되고 있으며, 많은 국내의 호환기종이 이것을 사용하고 있다. 최근에는 Phoenix BIOS를 탑재한 PC-AT 에서 AT용의 OS /2를 동작시킬 수도 있다고도 한다.

## ● 어댑터 보드의 클론

PC본체의 호환기 외에 확장보드의 클론도 있다. 이것은 IBM이 지원하는 정규의 확장보드와 바꿀가능을 보증하고, 또한 가격이 싸고 혹은 성능적으로 우수하다는 것이다. 또 1매의 확장보드에 표준보드 수매분의 기능을 집적한 다기능보드라고 불리우는 것도 있다. 이들의 호환확장보드는 IBM PC 혹은 지정한 호환기 위에서의 동작은 보증되지만 호환보드를 PC클론에 삽입한 경우 정확하게 동작하는 보증은 어디에도 없다.

최근의 경향으로서 재미있는 것은 어떤 특정한 호환보드가 IBM순정의 확장보드보다도 많이 보급되고 있으며, 사실상의 표준으로 되어 버린것도 있다. 특히 유명한 것은 Hercules Computer Technology의 허클레스 그래픽보드이다. 이것은 모노크롬 디스플레이어댑터(MDA)의 호환보드인 것이지만 기능적으로 순정품보다 우수하였다는 것과 프린터 인터페이스도 포함되어 있었기 때문에 압도적인 지지를 얻었던 것이다. 그 결과 확장그래픽스 인터페이스·어댑터(EGA)의 호환보드에서는 많은 보드가 허클레스모드를 지원하도록 되어 IBM 순정EGA는 거의 사용되지 않는다고 하는 것이 실상이다. 당연하지만 IBM순정의 EGA는 허클레스모드를 지원하지 않고 가격도 훨씬 높은 것이다.

# PC/2와 OS/2

PC AT는 IBM PC패밀리의 최상위 기종이라고 할 수 있는 것이다. 그러나 PC의 8비트 아키텍처나 그위의 오퍼레이팅 시스템을 도입할 수 없다.

IBM이 1987년에 발표한 PS/2시리즈는 앞으로의 IBM PC을 방향을 어느정도 명확하게 하였다. 즉 PC XT까지의 아키텍처 계승과 신아키텍처의 채용이다. PS/2-30은 종래의 PC XT와의 호환을 충분히 고려한 후계기종이며 내부의 16비트버스인 인터페이버스는 PC규격의 8비트·버스로 되어 있으며, XT의 소프트웨어, 하드웨어의 자산을 계승할수 있다. 그러나 PS/2-50, 60은 80286이 탑재된 AT의 후계기종이라고 볼 수 있지만 인터페이스버스의 호환성은 새삼스럽게 채용된 마이크로 채널(Micro channel)에 의하지 않게 되어 있다. 즉, PS/2-50 이상은 소프트웨어의 호환성이 있는 것과 PC AT와의 하드웨어버스의 호환성은 거의 고려되어 있지 않다.

그러한 의미로는 PC AT라고 하는 하드웨어는 이미 과거의 것으로 되었다고 볼 수 있지만 XT와의 높은 호환성과 클론메이커에 의한 값싼 하드웨어의 공급에 의해서 당분간은 계속해서 PC의 하나의 표준으로서 존재하리라고 생각된다. 표 5는 PS/2시리즈의 사양을 나타낸다.

OS의 면에서 보면 IBM PC/2-50 이상에서도 현재 표준으로 지원되는 OS는 MS-DOS3.3이다. 그러나 앞에서 설명한 OS/2가 옵션설정되어 있어서 80286/386의 프로텍트 모드가 지원되도록 되었다. 표 4에서도 알 수 있는 바와같이 OS/2는 본격적인 멀티태스크 OS임에도 불구하고 종래의 MS-DOS상의 응용프로그램도 실행할 수 있기 때문에 MS--DOS 에서의 이행이 쉬워서 서서히 보급되리라 생각되지만 아직 프로텍트모드를 사용한 응용 소프트웨어가 있는 것은 아니다.

OS/2를 본격적으로 사용한 소프트웨어가 시장에 투입되기까지에는 다소 시간이 걸린다고 하는 예상도 있다. 이것은 OS/2를 위한 그래픽스 인터페이스인 프레젠테이션 매니저(Presentation Manager)가 아직 공개되어 있지 않는(1988년 11월 발표 예정) 것도 관계한다.

〔표 5〕 PS/2시리즈의 주요 규격

| | PS/2model30 | PS/2model50 | PS/2model60 | PS/2model80 |
|---|---|---|---|---|
| CPU | 8086-2 | 80286 | 80286 | 80386 |
| 표준 메모리 | 640kB | 1MB | 1MB | 2MB |
| 확장 메모리 | — | 7MB까지 | 15M바이트까지 | 16MB까지 |
| 플로피 디스크와 그 용량 | 3.5인치×1<br>720kB | 3.5인치×1<br>1.44MB | 3.5인치×1<br>1.44MB | 3.5인치×1<br>1.44MB |
| 하드 디스크 | 20MB×1 | 20MB×1 | 44MB<br>70MB | 40MB<br>70MB<br>115MB |
| I/O 슬롯 | 3 | 3 | 7 | 7 |
| OS | PC-DOS3.3 | PC-DOS3.3<br>OS/2 | PC-DOS3.3<br>OS/2 | PC-DOS3.3<br>OS/2 |
| 비 고 | PC XT의 상위 기종.<br>XT 버스를 채용 교육용으로서 Model 125라는 것이 있다. | PC AT의 상위 기종에 상당하지만 AT버스를 채용하지 않고 마이크로 채널을 이용 | Model 150의 상위기종.<br>플로어 스탠드형으로 하드 디스크 용량과 I/O 슬롯을 증가시켰다. 마이크로 채널을 채용 | 80386을 탑재한 최상위 기종. 일본 IBM의 5570과 호환성이 있다. 마이크로 채널을 채용 |

프레젠테이션 매니저는 현재의 MS-Windows의 확장판이라고도 할 수 있는 것이며, 매킨토시 등의 유저 인터페이스 등과 비교되는 것이다. OS/2는 멀티태스크가 가능하기 때문에 종래의 ROM-BIOS에 의한 저레벨의 그래픽스조작은 자원의 공유라고 하는 점에서 극히 어렵게 되어 프레젠테이션 매니저를 경유하는 것을 필수로 된다.

PC/2, OS/2가 앞으로 어떻게 진화하는 가는 대단히 흥미가 있는 것이다. IBM이 하는 것이면 반드시 성공한다고 하는 것도 아니다. IBM PC시리즈에서도 PCjr이라든가 3270 PC라고 하는 실패가 있었던 것이다.

PS/2, OS/2에 관한 불안이 없는 것은 아니다. 그 하나는 OS/2와 80386의 관계이다. 즉 OS/2의 하드웨어 모델이 80286의 프로텍트 모드이며 80386을 전제로 하고 있지 않는 점이다. 마이크로소프트사의 노력에 따라서 OS/2에서 MS-DOS의 응용프로그램이 동작하도록되어 있는 것이지만 80386의 사용이 전제이면 이야기는 다르다.

80386에서는 가상8086모드가 있어서 여러개의 MS-DOS응용프로그램을 상위 OS의 관리하에서 멀티태스크로 움직이게 하는것이 가능하다. MS-DOS와의 호환성에 관한 한 80386의 쪽이 80286보다도 높다고 보는 견해도 있다. 또한 80386에서는 32비트 확장어드레스 모드에서 4G바이트의 공간을 선형공간으로서 사용할 수 있지만 이것은 80286에서는 지원되고 있지 않다. 이 2가지를 고려하면 80386이 표준 CPU로 되는 시점에서 현재의 OS/2와는 상당히 다른 OS(OS/3?)가 나올 가능성도 있는 것이다. 즉 OS/2는 MS-DOS가 8086에 묶여서 진화에 한계가 있었던것 같이 80286아키텍처에 묶여질 가능성이 있다.

## 〈용어해설〉

**●더미 명령(Dummy Instruction)**
실행하지 않을 것을 전제로 하여 프로그램 내에 삽입되어 있는 것으로만 그럴싸한 명령으로서 실행하더라도 아무런 동작도 하지 않는 명령이다. 프로그램을 변경하거나 수정할 때 프로그램의 일부에 이 명령을 넣어두면 편리하다.

**●램(Random Access Memory : RAM)**
마이크로 컴퓨터 등에 사용되는 IC 메모리로 플립플롭으로 구성된 메모리 셀(기억단위)을 랜덤 액세스시에 세트·리세트하여 정보를 읽어내거나 써넣을 수 있도록 된 메모리이다.
랜덤 액세스란 메모리의 임의의 어드레스에 기억되어 있는 데이터를 어드레스 지정에 의해 바로 읽어내고 써넣는 것을 말한다.
램은 플립플롭에 의한 반도체 메모리이기 때문에 램에 공급되는 전원이 끊어지면 램에 기록되어 있는 내용은 소실된다(휘발성 : Volatile). 이에 대해 전원이 끊어져도 메모리의 내용이 남아 있는 것을 불휘발성(Non-volatile) 메모리라고 한다.

**●롬(Read Only Memory : ROM)**
이것은 제조할 때에 변경할 필요가 없는 프로그램이나 데이터를 기록해 넣고 그것을 리드할 수만 있게 해 놓은 것으로 새로이 기록할 수 없는 메모리를 의미한다.

**●마스크 롬(Mask ROM)**
ROM의 한 종류로 제조 공정에서 실리콘 기판 위에 전자 회로를 만들 때 사진 기술을 이용하여 회로 패턴에 빛을 마스크(빛이 통하지 않도록 사이를 막는다)하여 메모리 셀의 0, 1을 정하는 것으로 ROM이 완성될 때는 0, 1로 구성된 기계어 프로그램이 기록되어 있으며 리라이트가 불가능하다.
메모리 셀은 1개의 트랜지스터로 만들 수 있으므로 집적밀도가 높고 대용량의 칩이 가능하다.

**●모뎀(MOdulato- DEModulator : MODEM)**
단말 장치로부터의 직류 신호를 교류 신호로 변환(변조)하거나 CPU로부터의 교류 신호를 직류 신호로 변환(복조)하는 장치이다. 변복조 장치는 단말 장치측뿐만 아니라 중앙의 컴퓨터측에도 필요하다.

**●회송 시간(Turnaround Time)**
컴퓨터에 데이터나 프로그램이 입력되어 그에 대한 적절한 처리를 하고 그 결과를 얻을 때까지의 시간을 말하며, 반환 시간이라고도 한다.

**●오퍼레이팅 시스템(Operating System)**
프로그램 작성에 드는 노력이나 시간 또는 조작 등의 시간을 줄이고, 사람, 프로그램, 하드웨어 등의 컴퓨터 자원을 유효하게 이용할 수 있도록 준비하는 소프트웨어들의 집합이다.

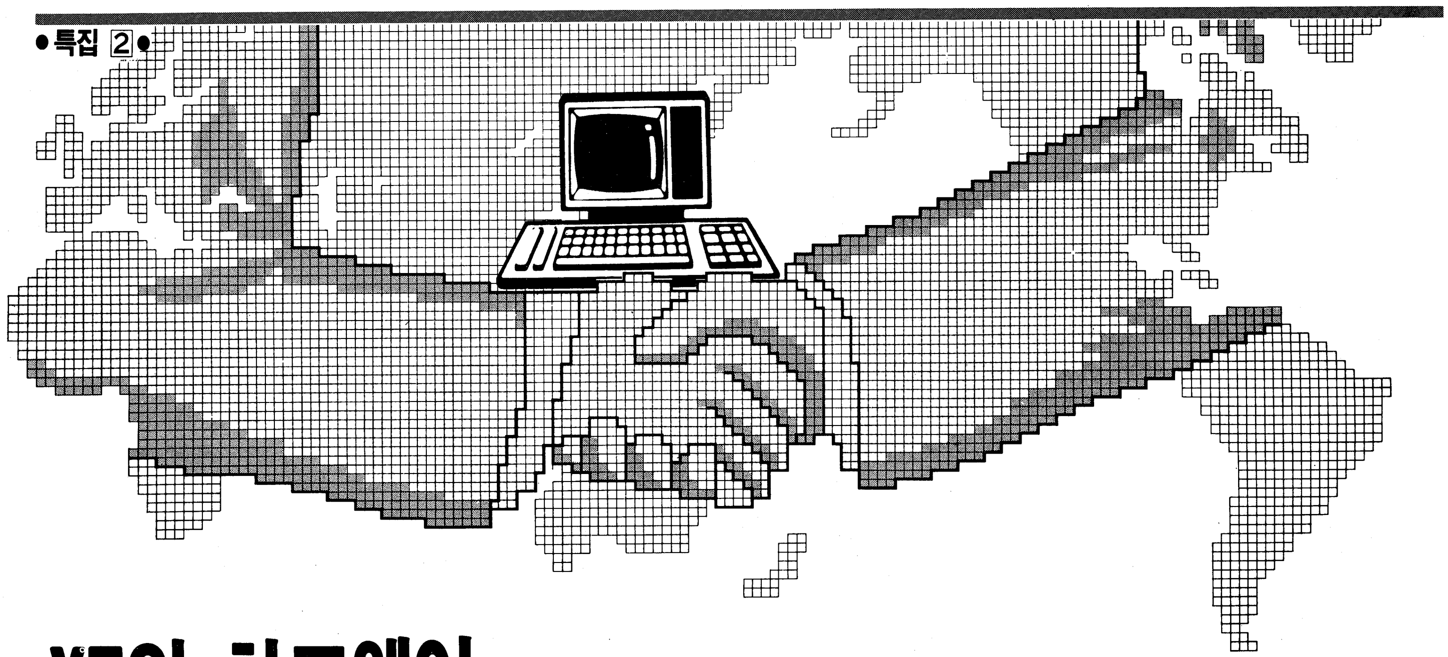
**●도큐멘테이션(Documentation)**
정보의 수집, 정리, 문서화, 보존, 배포 등의 정보 관리에 관한 기록을 말한다. 일반적으로 시스템 설계에 관한 서류나 프로그램의 조작설명서 등과 같이 시스템 전반의 약속을 문서화한 것이다.
이러한 기록들은 시스템 엔지니어와 프로그래머 및 사용자가 상호 연락이나 확인을 할 수 있게 하여 사용이나 변경시에 중요한 역할을 한다.

**●시스템 분석(System Analysis)**
문제의 제기에 따라서 현재의 시스템을 조사 분석하여(현상 분석), 이들의 해결을 위한 컴퓨터의 성능이나 경제성 등의 제약 또는 사무 담당자의 요구를 정리하여 시스템의 대상 범위를 명확히 하고 실현 가능한 가장 적합한 시스템안(기본 설계)을 결정하도록 지원하는 것을 말한다.

**●뱅킹 시스템(Banking System)**
은행의 온라인 실시간 시스템에서 계산 센터의 대형 컴퓨터를 써서 계산 처리를 하고 각 은행간에 공통 네트워크 시스템을 사용하여 고객에 대한 서비스를 하는 시스템이다.

**●리모트 배치처리방식(Remote Batch Processing System)**
먼 지점으로부터 중앙의 컴퓨터에 데이터를 입력하고 중앙의 컴퓨터로 일괄 처리하여 그 결과를 출력하는 방식.
즉흥성을 요하지 않는 처리를 먼 지점으로부터 이루고 싶은 경우에 사용한다. 리모트 작업 엔트리라고도 한다.

**●CAI용 언어(Computer Assisted Instruction Language)**
CAI용 언어는 교재나 학습용 프로그램을 바탕으로 하여 단말을 통해 학습자가 컴퓨터와 대화하는 방식으로 학습하는 교육 시스템을 쉽게 작성하도록 하는 전용의 프로그래밍 언어이다.



# XT의 하드웨어

현재의 XT는 초기 IBM PC와 그 설계사상에 큰 차이가 없다. 단지 XT에는 하드디스크의 지원이 있어서 확장보드의 I/O 슬롯이 5개에서 8개로 증가한 것, 카세트 인터페이스를 없게 하였다고 하는 변경점이 있을 뿐이다.

PS/2가 발표되어 AT의 후계기는 사실상 없게 되어 버렸지만 XT는 PS/2 모델30으로 되어 계속되고 있다. 즉 CPU는 8086으로 되어 있지만 XT용으로 개발된 확장보드나 소프트웨어는 모두 이용할 수 있도록 되어 있다.

## ●XT의 구성

IBM PC XT는 하드웨어적으로 보면 CPU에 8088을 사용한 16비트 머신이며 본체는 기본적으로 마더보드와 전원부로 구성되어 있다. 물론 이것만 가지고는 컴퓨터로서 동작하지는 않는다. 이것에 키보드, 각종 주변보드류, 플로피디스크, CRT 디스플레이, 프린터 등의 주변장치가 필요하다.

표 1에 IBM 표준의 부품 리스트를 나타낸다.

즉 유저마다 사용목적이 현저하게 다른 컴퓨터에서는 옵션품으로서 분할할 수 있는 것은 될수록 나누어 유저의 요망에 따라서 구성을 변화할 수 있도록 하는 것이 좋다고 하

〔표 1〕 IBM PC XT/AT의 순 옵션품 리스트

| | |
|---|---|
| IBM Expantion Unit······XT만(확장 박스) | |
| IBM Monochrome Display··········XT/AT<br>IBM Color Display····················XT/AT<br>IBM Enhanced Color Display······XT/AT | CRT 디스플레이 |
| IBM Graphics Printer······XT/AT<br>IBM Personal Computer Color Printer······ XT/AT<br>IBM PC Compact Printer······XT/AT | 프린터장치 |
| IBM 5.25" Diskette Drive··········XT만<br>IBM Slimline Diskette Drive······XT만<br>IBM 10MB Fixed Disk Drive·····XT만<br>IBM PC AT High Capacity Diskette Drive······AT만<br>IBM PC AT Double Sided Diskette Drive ·······AT만<br>IBM PC AT 20MB Fixed Disk Drive··············AT만 | 플로피 및 하드 디스크 |
| IBM 64/256KB Memory Expansion Option and IBM 64KB Memory Module Kit······XT만<br>IBM PC AT 128KB Memory Expansion Option······AT만<br>IBM PC AT 512KB Memory Expansion Option······AT만 | 확장 메모리 보드 |
| IBM Monochrome Display and Printer Adapter······XT/AT<br>IBM Color/Graphics Monitor Adapter··················XT/AT<br>IBM Enhanced Graphics Adapter··························XT/AT<br>IBM Printer Adapter·········································XT만<br>IBM 5.25" Diskette Drive Adapter························XT만<br>IBM Fixed Disk Adapter···································XT만<br>IBM PC AT Fixed and Diskette Adapter···············AT만<br>IBM Asynchronous Communication Adapter ··········XT만<br>IBM Synchronous Data Link Control (SDLC)Communications Adapter····XT/AT | I/O 인터페이스 보드 |
| IBM Cluster Adapter·························XT만<br>IBM Game Control Adapter··········· XT/AT | |
| IBM Prototype Card·······················XT만<br>IBM PC AT Prototype Adapter······AT만 | |

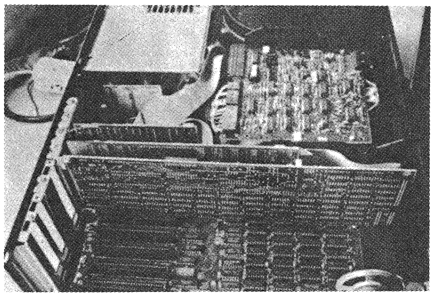
는 사고방법인 것이다. 예를 들면 최근에 나오는 일부기종은 컬러그래픽이 최초부터 부착되어 있다. 그것도 최신LSI를 사용한 아날로그RGB 보드 등으로 더할 나위 없지만 모노크롬 디스플레이로 사용하는데 충분한 유저로서는 그러한 것은 필요하지 않는 것이 된다. IBM PC시리즈에서는 그래픽보드도 해상도를 따라서 수종류 있고, 시리얼 / 패러렐 보드와 디스크관계의 I/O인터페이스도 여러종류가 옵션으로 되어 있다. 이것은 각각의 유저 희망에 의해서 여러가지의 시스템으로 구성할 수 있도록 되어있는 것이다.

옵션보드는 물론 나중에 갱신 혹은 추가가 가능하므로 예산이나 사용상황에 따라서 기능향상이 가능하다.

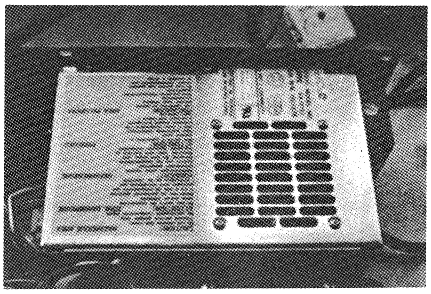
이와 같은 시스템은 어느 정도 생명이 길지 않아서는 안된다. 반년 혹은 1년에 다음의 기종이 발매되고, 더욱 조금씩 스펙이 변화하고 있다면 유저는 불안에 견디지 못한다.

최근의 시스템구성에서는 대개 CRT콘트롤러와 플로피디스크 콘트롤러를 포함하고 있지만 IBM PC에서는 옵션품으로 되어 있다.

〔사진 1〕 PC XT의 내부 형태



〔사진 2〕 PC XT의 전원



## • XT의 전원

PC XT의 전원은 130W의 용량이고 기본적으로 4종류의 직류전압을 공급한다. 전원부의 간단한 사양을 표 2에 나타낸다. 공급선은 메인보드, FDD, HDD등이다. 기타 파워·굿신호라고 하는 것이 나오고 있으며, AC DC 양전원 모두 소정의 전압이상이 아니면 인터럽트를 발생하도록 하는 구조로 되어 있다. AC전원은 110V(50/60Hz)입력이며 전원전압 범위가 90~137V로 되어 있으므로 국내 환경에서도 그대로 동작한다.

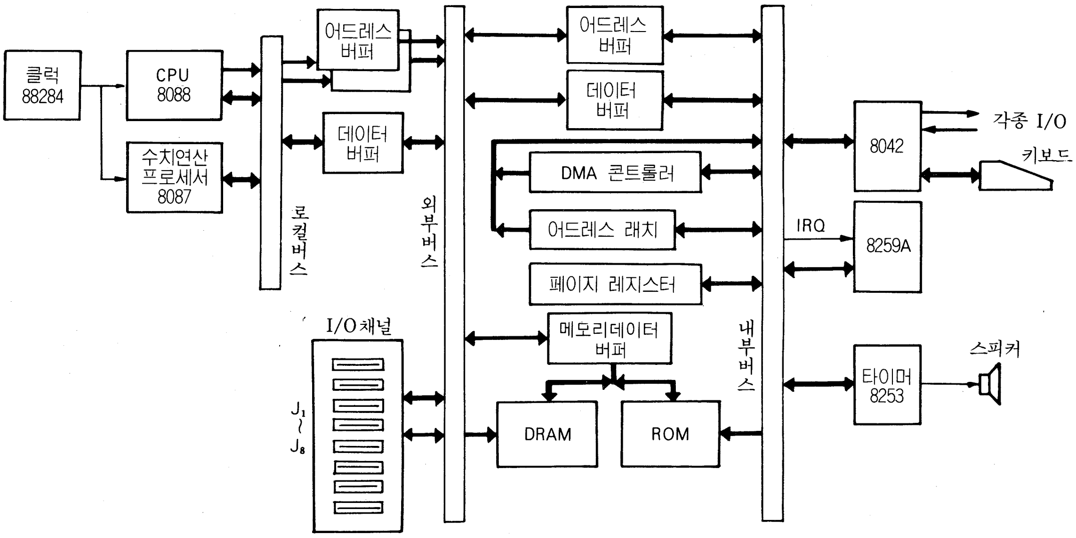
## • XT의 시스템보드

IBM PC XT의 블록도를 그림 1에 나타낸다. 시스템 보드에는 2종류가 있으며, 메모리용량이 최대256K바이트의 것과 640K바이트의 것이 있다. 각각 64/256K, 256/640K시스템 보드라고 불리우고 있다. 이 명칭은 최소 / 최대 탑재메모리용량을 나타내고 있다. 이들의 차이는 메모리용량뿐이며 그 외의 구성은 전부 같다.

## • XT의 시스템버스

시스템보드의 버스는 기본적으로 CPU를 중심으로 한(로컬버스), RAM과 내부 I/O슬롯용(내부버스라 한다), 내부의 주변LSI를 위한 I/O와 확장유니트용(외부버스)의 3개로 구성되어 있다. 로컬버스는 DMA사이클에 들어 가면 기타의 버

〔그림 1〕 IBM PC XT의 블록도



〔표 2〕 PC XT의 전원규격

| 입력 |
|---|
| 50/60Hz ±3Hz<br>90V(min), 110V(typ),137V(max)<br>180V(min), 220/240V(typ),259V(max) |
| DC출력 |
| +5V, 2.3~15.0A<br>−5V, 0~ 0.3A<br>+12V, 0.4~ 4.2A<br>−12V, 0~ 0.25A |
| AC출력 |
| 120V , 0~1.0A<br>220/240V, 0~0.5A |

〔표 3〕 PC XT의 메모리 맵

| 어드레스 | 기 능 | |
|---|---|---|
| 00000~9FFFF | 시스템 메모리 | RAM영역 |
| B0000~B7FFF | MDA(모노크롬) | RAM영역 |
| B8000~B8FFF | CGA(컬러) | RAM영역 |
| C0000~C5FFF | EGA(인헨스트) | RAM영역 |
| C6000~C7FFF | PGA(프로페셔널) | RAM영역 |
| C8000~CBFFF | 하드 디스크 | RAM영역 |
| CC000~CFFFF | PC네트워크 | RAM영역 |
| D0000~D3FFF | 클러스터 | RAM영역 |
| D4000~EFFFF | 192kB 확장 ROM | ROM영역 |
| F0000~FFFFF | 64KB BIOS, Basic ROM | ROM영역 |

스에서 분리되지 않으면 안되고, 내부버스(확장코넥터 7슬롯)를 드라이브하지 않으면 안된다. 이 때문에 로컬버스와 내부버스의 사이에 버스버퍼가 놓여진다.

단, 내부버스에 있는 DRAM은 64K 또는 256K의 것이 최대 4뱅크로 구성되어 있으므로 DRAM전용의 버퍼가 마련되어 있다. 또 외부버스는 내부버스에서 분기한 버스이며 여기에 시스템보드내의 주변LSI와 ROM, 확장 유니트용 슬롯이 매달린다. 내부버스가 이들의 모두를 드라이브하는 것은 상당히 난처하기 때문에 여기에도 버스버퍼가 들어 간다.

## ● XT의 메모리 맵

시스템구성을 알기 위한 또 하나의 방법은 각 디바이스의 어드레스 할당을 조사하는 것이다. XT의 메모리 맵을 표 3에 나타낸다. I/O슬롯에 어댑터보드를 꽂고 있지 않을 때 메모리 맵상에 존재하는 것은 RAM 과 ROM뿐이다.

DRAM은 64/256K시스템이 64K비트×9×4뱅크, 256/640K시스템이 256K비트×9×4뱅크로 구성되어 있다. RAM으로서 사용 할 수 있는 영역의 최대는 640K바이트이며, MS-DOS 머신의 전형적인 메모리 구성이라고 하기 보다 다른 MS-DOS 머신이 모두 IBM PC로 되었다고 하는 쪽이 옳을지도 모르겠다. ROM 영역은 어드레스공간의 상위에 있으며 리세트 벡터를 포함하는 F0000H에서의 64K바이트에 BIOS-ROM이 배치되어 있다. BIOS-ROM에는 POST(Power On Self Test), I/O드라이버, 부트스트랩 로더 등 시스템가동에 필요한 최저한의 루틴이 수납되어 있다.

## ● XT의 I/O맵

PC XT의 I/O맵은 표 4와 같이 기본적으로 2개로 분할되어 배치되어 있다. 하위에 있는 것은 시스템보드상에 있는 주변LSI와 TTL조직에 의한 I/O용, 또 하나의 I/O슬롯에 꽂는 어댑터 보드용의 것이다. 시스템보드상의 주변LSI는 매뉴얼대로의 선택이 이루어지고 있고, 그 사용법도 매뉴얼에 준한 것으로 되어 있지만 TTL래치를 부가하여 1M바이트까지의 페이지액세스를 가능하게 하고 있다. 8237A는 1개뿐이므로 DMA는 최대 4채널이라는 것으로 된다. 단, 채널0은 DRAM의 리플레시에 사용되므로 나머지 3채널이 DMA전송용으로서 개방되어 있다.

XT의 하드웨어 인터럽트의 요인 일람을 표 5에 나타낸다. IBM PC시리즈의 NMI는 무슨이유인지 마스크가능으로 디자인되어 있어서 I/O 어드레스상의 NMI마스크 레지스터로 제어할 수 있다. IRQ는 8259A에 의해서 8레벨까지 서포트되어 있지만 $IRQ_0$는 타이머인터럽트로 날짜 및 시간의 갱신에, $IRQ_1$은 키보드인터럽트로 각각 시스템 보드에서 사용되고 있다.

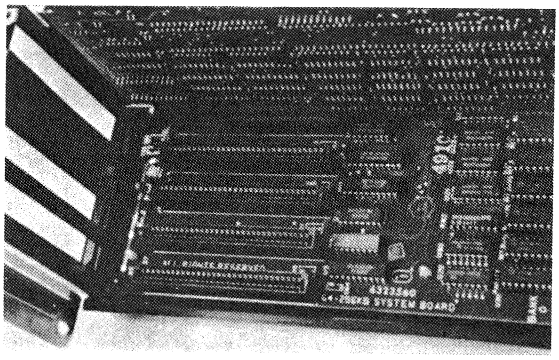
〔표 4〕 PC XT의 I/O 맵

| I/O 어드레스 | 디바이스 |
|---|---|
| 000~01FH | DMA콘트롤러 8237A |
| 020~03FH | 인터럽트 콘트롤러 8259A |
| 040~05FH | 타이머 8253 |
| 060~06FH | PPI 8255A |
| 080~09FH | DMA 페이지 레지스터 |
| 0AXH | NMI 마스크 레지스터 |
| 200~20FH | 게임 콘트롤 |
| 201H | 게임 I/O |
| 210~217H | 확장 유니트 |
| 278~27FH | 패러럴 프린터 2 |
| 2B0~2DFH | EGA |
| 2E1H | GPIB |
| 2E2, 2E3H | 데이터영역 |
| 2F8~2FFH | 리얼 포트 2 |
| 300~31FH | 프로토 타입 카드 |
| 320~32FH | 하드 디스크 |
| 360~36FH | PC 네트워크 |
| 378~37FH | 패러럴 프린터 1 |
| 380~38FH | SDLC, BSC 2 |
| 390~393H | 클러스터 |
| 3A0~3AFH | BSC 1 |
| 3B0~3BFH | MDPA(모노크롬, 프린터) |
| 3C0~3CFH | EGA |
| 3D0~3DFH | CGA |
| 3F0~3F7H | 디스크 콘트롤러 |
| 3F8~3FFH | 리얼포트 1 |

〔표 5〕 인터럽트 원인

| 인터럽트 | 요 인 |
|---|---|
| NMI | 메모리 패리티 에러 또는 8087 |
| $IRQ_0$ | 타이머 |
| $IRQ_1$ | 키보드 |
| $IRQ_2$ | EGA, PC 네트워크 |
| $IRQ_3$ | 비동기통신 2, 기타 |
| $IRQ_4$ | 비동기통신 1, 기타 |
| $IRQ_5$ | 하드 디스크 |
| $IRQ_6$ | 플로피 |
| $IRQ_7$ | 프린터 |

〔사진 3〕 PS/2 MODEL 30



따라서 나머지 6개를 다른 I/O디바이스로 사용할 수 있다. 그러나 이들은 표5와 같이 이미 예약되어 있어서 자작의 유니버설 보드에서의 인터럽트처리가 필요한 경우는 주의가 필요하다. 또 시스템 중에서 가장 강력한 인터럽트는 CPU리세트이지만 XT에는 매뉴얼 리세트를 위한 회로가 없다.

시스템 타이머 8253은 메모리 리플레시요구(Clock 1), 날짜/시간갱신을 위한 인터럽트(Clock 0), 그리고 스피커음원(Clock 2)에 사용되고 있다. 그러므로 유저가 프로그램할 수 있는 것은 스피커음원만에 한정된다. 또 타이머의 클럭입력주파수는 전채널 모두 1.193182MHZ이다.

패러럴 보드 8255A의 주 역할은 키보드I/O와 시스템구성정보의 입출력이다. 표6에 각 I/O포트의 기능을 나타낸다. 또 키보드와 본체는 9비트 프레임의 직렬통신에서 정보의 주고받음으로 8255A에는 그 병렬I/O를 직렬로 변환하기 위한 TTL 시프트 레지스터가 접속되어 있다.

## • XT의 I/O슬롯

PC XT의 I/O슬롯은 합계8개 있는 것이지만 그 중의 3개는(초기 IBM PC의 I/O슬롯에 추가된 분), 짧은 칫수의 어댑터보드만 실장가능하다.

I/O슬롯은 $J_1$~$J_7$까지가 어댑터보드용이며, $J_8$코넥터는 확장유니트 전용으로 되어 있다. 그림2에 I/O슬롯의 핀 배열을 나타낸다. 각 핀의 기능은 AT의 62핀 코넥터와 같다.

〔표 6〕 PC XT의 I/O 포트(8255A)의 기능

| I/O어드레스 | 포트번호 | 기 능 |
|---|---|---|
| 060H(입력) | $PA_0$~$PA_7$ | 키보드 스캔 코드 또는 진단 출력 |
| 061H(출력) | $PB_0$ | 타이머 게이트 2 (스피커) |
| | $PB_1$ | 스피커 데이터 |
| | $PB_2$ | 예 비 |
| | $PB_3$ | 딥 스위치 선택 |
| | $PB_4$ | RAM 패리티 체크 인에이블 |
| | $PB_5$ | I/O채널 체크 인에이블 |
| | $PB_6$ | 키보드 클럭 홀드 |
| | $PB_7$ | 키보드 클리어 |
| 062H(입력) | $PC_0$ | POST 루프($SW_1$), 디스플레이 0($SW_5$)*2 |
| | $PC_1$ | 8087 인스톨($SW_2$), 디스플레이 1($SW_6$)*2 |
| | $PC_2$ | RAM 크기($SW_3$)*1, 드스크 드라이브($SW_7$)*3 |
| | $PC_3$ | RAM 크기($SW_4$)*1, 디스크 드라이브($SW_8$)*3 |
| | $PC_4$ | 예 비 |
| | $PC_5$ | 타이머 채널 2출력 |
| | $PC_6$ | I/O채널 체크 |
| | $PC_7$ | RAM패리티 체크 |

*1 메모리용량의 선택

| $SW_4$ | $SW_3$ | 시스템 64/256k | 시스템 256/640k |
|---|---|---|---|
| 0 | 0 | 64K | 256K |
| 0 | 1 | 128K | 512K |
| 1 | 0 | 192K | 576K |
| 1 | 1 | 256K | 640K |

*2 디스플레이의 선택

| $SW_6$ | $SW_5$ | 디스플레이 타 입 |
|---|---|---|
| 0 | 0 | 예약 |
| 0 | 1 | 컬러—40×25 |
| 1 | 0 | 컬러—80×25 |
| 1 | 1 | 모노크롬80×25 |

*3 드라이브 대수의 선택

| $SW_8$ | $SW_7$ | 디스크 드라이브 대 수 |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 2 |
| 1 | 0 | 3 |
| 1 | 1 | 4 |

〔그림 2〕 I/O 슬롯의 핀배열(I, O, I/O는 마더보드에서 본 경우의 입력, 출력 또는 쌍방향을 의미한다.)

리어 패널측

| | | | | | |
|---|---|---|---|---|---|
| 그라운드 | GND | $B_1$ | $A_1$ | −I/O CH CK | I |
| O | RESET DRV | $B_2$ | $A_2$ | $SD_7$ | I/O |
| 전원 | $+5V_{DC}$ | $B_3$ | $A_3$ | $SD_6$ | I/O |
| I | $IRQ_2$ | $B_4$ | $A_4$ | $SD_5$ | I/O |
| 전원 | $-5V_{DC}$ | $B_5$ | $A_5$ | $SD_4$ | I/O |
| I | $DRQ_2$ | $B_6$ | $A_6$ | $SD_3$ | I/O |
| 전원 | $-12V_{DC}$ | $B_7$ | $A_7$ | $SD_2$ | I/O |
| I | −CARD SLCTD | $B_8$ | $A_8$ | $SD_1$ | I/O |
| 전원 | $+12V_{DC}$ | $B_9$ | $A_9$ | $SD_0$ | I/O |
| 그라운드 | GND | $B_{10}$ | $A_{10}$ | I/O CH RDY | I |
| O | −MEMW | $B_{11}$ | $A_{11}$ | AEN | O |
| O | −MEMR | $B_{12}$ | $A_{12}$ | $SA_{19}$ | I/O |
| I/O | −IOW | $B_{13}$ | $A_{13}$ | $SA_{18}$ | I/O |
| I/O | −IOR | $B_{14}$ | $A_{14}$ | $SA_{17}$ | I/O |
| O | $-DACK_3$ | $B_{15}$ | $A_{15}$ | $SA_{16}$ | I/O |
| I | $DRQ_3$ | $B_{16}$ | $A_{16}$ | $SA_{15}$ | I/O |
| O | $-DACK_1$ | $B_{17}$ | $A_{17}$ | $SA_{14}$ | I/O |
| I | $DRQ_1$ | $B_{18}$ | $A_{18}$ | $SA_{13}$ | I/O |
| I/O | $-DACK_0$ | $B_{19}$ | $A_{19}$ | $SA_{12}$ | I/O |
| O | CLK | $B_{20}$ | $A_{20}$ | $SA_{11}$ | I/O |
| I | $IRQ_7$ | $B_{21}$ | $A_{21}$ | $SA_{10}$ | I/O |
| I | $IRQ_6$ | $B_{22}$ | $A_{22}$ | $SA_9$ | I/O |
| I | $IRQ_5$ | $B_{23}$ | $A_{23}$ | $SA_8$ | I/O |
| I | $IRQ_4$ | $B_{24}$ | $A_{24}$ | $SA_7$ | I/O |
| I | $IRQ_3$ | $B_{25}$ | $A_{25}$ | $SA_6$ | I/O |
| O | $-DACK_2$ | $B_{26}$ | $A_{26}$ | $SA_5$ | I/O |
| O | T/C | $B_{27}$ | $A_{27}$ | $SA_4$ | I/O |
| O | ALE | $B_{28}$ | $A_{28}$ | $SA_3$ | I/O |
| 전원 | $+5V_{DC}$ | $B_{29}$ | $A_{29}$ | $SA_2$ | I/O |
| O | OSC | $B_{30}$ | $A_{30}$ | $SA_1$ | I/O |
| 그라운드 | GND | $B_{31}$ | $A_{31}$ | $SA_0$ | I/O |

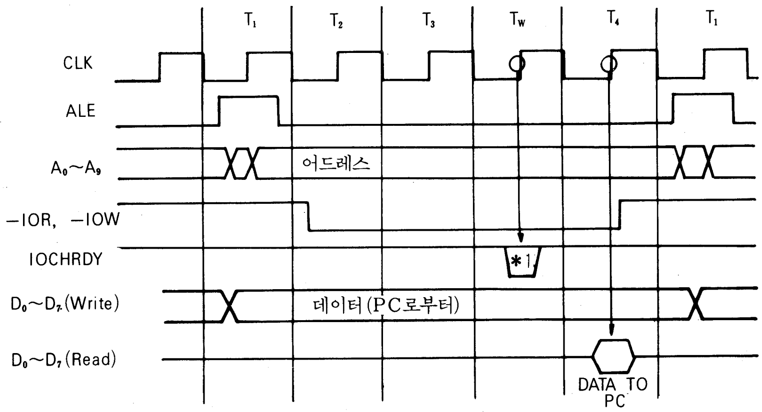
(전원 이외의 −는 부논리를 나타낸다)

프론트 패널측

그림 3에는 62핀 코넥터의 핀 배열에 의거한 입출력시의 타이밍도를 나타낸다. 여기에서 보면 입출력 신호가 지연될 경우 $T_w$사이클의 CLK 위상에 동기로서 IOCHRDY를 "L"로 하므로써 웨이트사이클을 추가할 수 있음을 알 수 있다. 여기서 웨이트사이클은 PC 내부에서 자동적으로 삽입된다.

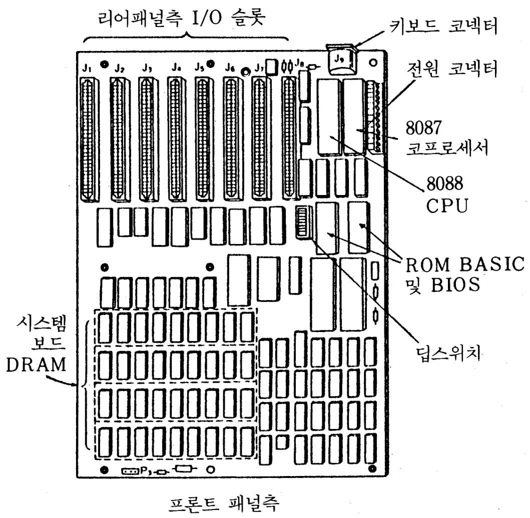
그림 4에는 PC XT의 시스템 보드 배치도를 예로써 보인다.

〔그림 3〕 IBM PC의 62핀 I/O채널의 기본 타이밍




(*1) 지연 입출력장치의 경우는 $T_w$ 사이클의 CLK 위상에 동기로서 IOC HRDY를 "L"로 하는 것에 의해서 웨이트 사이클을 추가할 수 있다. 1Wait Cycle는 PC 내부에서 자동적으로 삽입된다.

〔그림 4〕 IBM PC XT 시스템보드 배치도



●컬럼A●

# 오픈 아키텍처란

수년전의 신문이나 잡지에 소개된 퍼스컴 관계의 해외뉴스에는 IBM 사와 애플컴퓨터사가 치열한 시장획득경쟁을 전개하고 있으며 IBM이 애플을 역전하였다고 하는 것이 큰 화제거리였다.

IBM PC시리즈의 하드웨어는 같은 성능과 가격을 목표로 한 타사의 퍼스컴과 비교하여 특별히 우수하다는 것은 눈에 띄지 않는다. 오히려 애플컴퓨터사가 전력을 다하여 개발한 매킨토시 등에 비하면 뒤떨어지는 면이 많을 것이다.

미국에서 IBM PC의 시장점유율이 늘어난 이유의 하나로서 IBM이 IBM PC시리즈의 모두에 오픈 아키텍처를 채용하여 온 것을 들 수 있다. 즉 시스템 아키텍처의 정보를 공개하여, 회로도에서 BIOS의 소스리스트까지 바인더가 붙은 매뉴얼로 공급되고 있다. 물론 이것은 옵션품이지만 필요하면 누구라도 살 수 있기 때문에 그렇게 되면 IBM PC의 완전카피도 만들 수 있다. 단, 이것은 영리목적이 되면 저작권법위반이라는 단서가 붙지만.

유저로서는 응용 소프트웨어가 소정의 동작을 하여 결과가 같으면 되는 것이며, 퍼스컴은 단순한 블랙박스로서의 역할밖에 하지 않는다. 그래서 전혀 다른 BIOS에서 결과가 같은 것을 만드므로써 IBM PC의 클론을 실현할 수 있다. 요는 IBM의 BIOS를 전부 그대로 카피하고 있지 않으면 되는 것이다.

●컬럼B●

# 인텔과 모토롤러의 전쟁

## ●CPU인가 MPU인가

IBM PC의 하드웨어는 CPU에 인텔8088을 사용하고 플로피디스크는 표준탑재하고 있으며 IPL에 의해서 DOS가 언제라도 사용할 수 있도록 설계되어 있었다. 이것은 대단히 중요하며 각종 응용 소프트웨어가 공통의 토양위에서 성장하기 위해서는 절대 필요한 것이다. 또 IBM PC는 8비트 버스이기는 하지만 여기서 개발한 많은 소프트웨어 자원은 거의 변경없이 16비트 버스(8086, 80286)로 이행가능한 프로세서를 선택하고 있었던 것도 빠뜨릴 수 없을 것이다.

이 당시 마이크로프로세서 전쟁은 인텔과 모토롤러가 과격한 불꽃을 튀겨서 우리들도 다음의 마이크로프로세서가 어떤 것으로 될 것인가 대해 한시도 눈을 뗄 수 없는 상황이였다. IBM PC가 발매되기 조금 전 인텔은 8080, 8085에서 8088/86으로, 모토롤러는 MC6800에서 MC6809로 주력을 바꾸었다. 8088/86은 MC6809에 다소 뒤진 것이지만 16비트 대응이며 그 후의 OA용으로 많이 사용되게 되었다.

CPU의 기능확장은 데이터어장이나 스피드만은 아니다. 이 당시에는 이미 소프트웨어측의 요구로서 메모리공간이 64K바이트로는 충분하지 않게 되어 있었다. 그래서 모토롤러는 MMU(메모리 매니지먼트시스템)를 외부에 첨가하는 방법으로 인텔은 세그먼트관리에 의한 64K바이트단위의 메모리관리에 의해서 그 요구를 실현하였다. 바꾸어 말하면 모토롤러는 하드웨어적으로 인텔은 소프트웨어적으로 이들의 요구에 따른 것이다. 이 특징은 각각의 대표적인 OS에 현저하게 나타나고 있다.

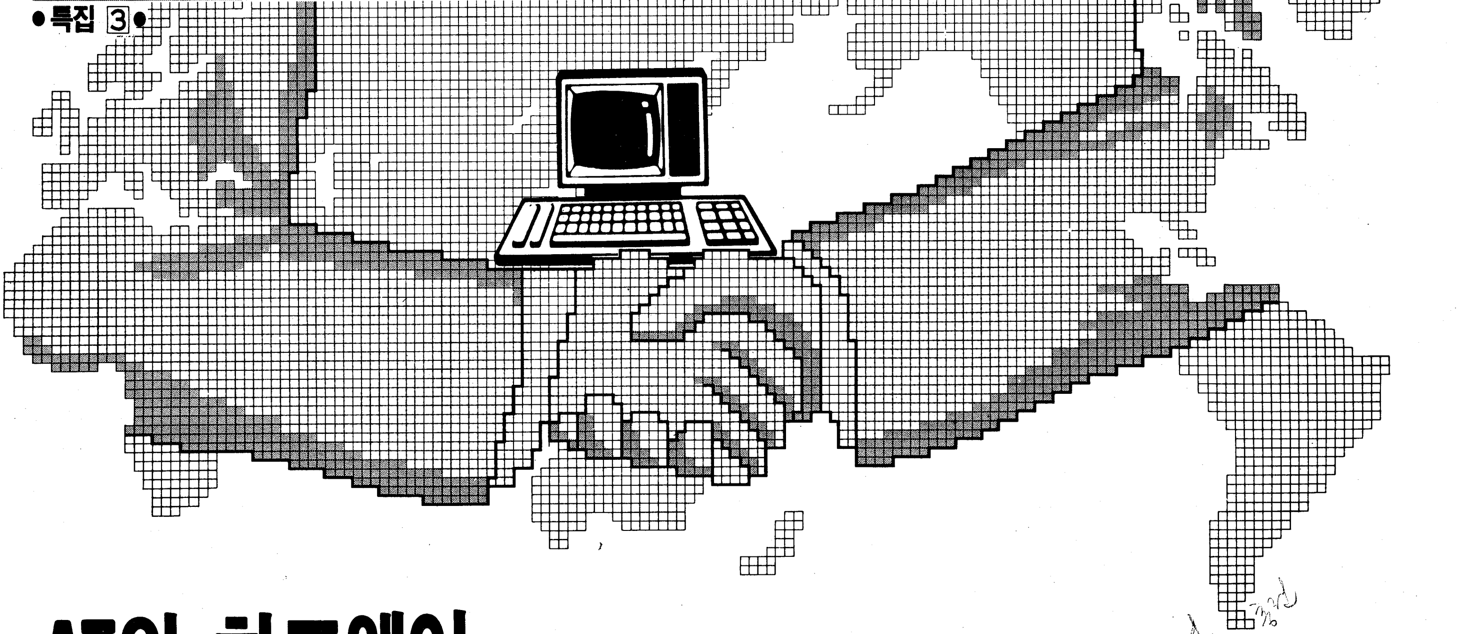
## • CPU에서 OS로

68계 시스템은 OS-9에 의해서 일찍부터 멀티유저, 멀티태스크의 혜택를 받을 수 있었지만 이것에 대해서 80계는 OS에 의한 메모리관리가 충분히 이루어지고 있지 않는 MS-DOS에서는 싱글태스크, 싱글유저로 되어 버린다. 단, 8비트 CPU에 있어서 멀티태스크, 멀티유저의 OS는 확실히 무거운 부하로 되어 MC6809에 의한 OS-9는 일부의 열광적 호비스트에게만 사용되고, 그 훌륭함은 MC68000으로 되어 비로소 진가를 발휘하게 되었다. 현실적으로는 여러가지 복잡한 처리를 능숙하게 사용하지 않으면 안되는 공작기계의 제어 등에 위력을 발휘하고 있다.

이것에 대해서 유닉스의 환경을 도입하고 있다고 할 수 있는 싱글태스크, 싱글유저의 MS-DOS가 왜인지 가장 많은 유저를 떠맡고 성장하여 왔다. 이것은 비지니스 머신으로서의 역할 때문이라고도 생각된다. 비지니스 머신으로서 중요한 것은 필요한 정보가 곧 얻어지는(데이터베이스) 즉, 자기가 생각했던대로 정보의 가공이 가능(워드프로세서, 스프레드쉬트 등), 이들의 처리를 위한 조작을 특별한 교육없이 할 수있는 환경 등일 것이다.

즉 일반 사용자들이 별 무리없이 사용할 수 있는 조작환경을 갖추는 것이 선결이며, 처리능력에 대해서는 그들의 요구가 고도로 되고 나서 대응하여도 늦지 않다. 그러므로 복잡한 멀티태스크 OS를 초기의 시점에서 투입하여도 컴퓨터의 보급효과는 높지 않다.

일반 유저가 사용하는 비지니스응용의 퍼스컴이 멀티태스크가 아니면 안되는 이유는 그다지 눈에 띄지 않는다. 강하게 그와 같은 응용이 있다고 하면 POS시스템과 같은 온라인처리를 수반한 것이며, 이렇게 되면 각종 단말이나 통신관계를 포함하여 설비투자가 이루어져야 한다.



# AT의 하드웨어

PC XT에 이어서 PC AT가 등장한 것은 1984년이다. IBM은 8088에서 단숨에 80286으로 차기 주력머신의 CPU를 바꾸어 버린것이다.

이 결단의 이유는 명백하지는 않지만 AT의 프로젝트를 시작한 시기에 와서 처음으로 IBM이 마이크로시장이 중요하다고 인식한 것인지도 모르겠다.

IBM은 PC AT를 등장시키면서 소프트웨어의 호환성을 꾀한다고 하는 IBM의 정신을 그대로 지속시켰다. 즉 대부분의 기존의 PC XT용의 소프트웨어가 변경없이 PC AT에서도 동작하도록 한 것이다. 이것은 80286이라는 CPU가 프로텍트모드와 리얼모드라고 하는 2개의 모드를 가지며, 리얼모드에서는 8088 / 8086의 명령을 완전히 지원하고 있는 것에 의하고 있는 것이지만 CPU가 바뀌어도 종래의 소프트웨어를 사용할 수 있다고 하는 점은 중요한 것이다.

기타 호환성을 유지한다고 하는 점에서 필요한 조건은

- I/O 관계의 어드레스에 변경이 없을 것
- I/O 디바이스 드라이버의 콜링 시퀀스와 기능이 동일할 것

등이 필요하다. 물론 PC AT에서는 이들을 완전히 만족시키고 있지만 이것만으로는 신상품으로서의 가치가 희박하므로 기타 여러가지의 확장이 시행되었다. 주된 특징은

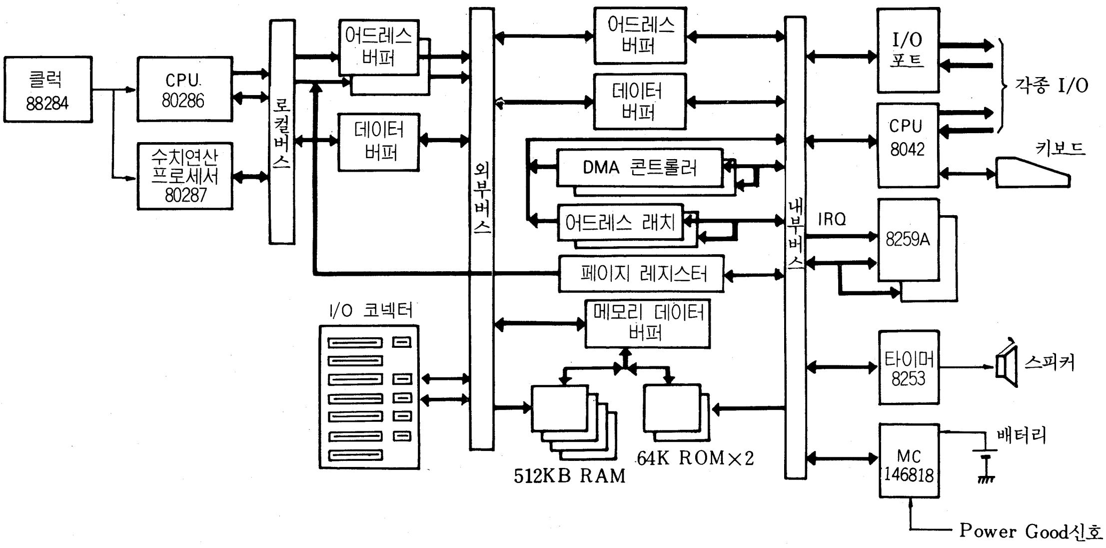
- CPU에 80826을 채용하였다.
- 시스템 지원기능을 충실히 하였다.
  - (a) 7채널의 DMA를 마련
  - (b) 16레벨의 인터럽트로 확장
  - (c) 시스템 클럭내장
  - (d) 프로그래머블 타이머내장
- 64바이트 ROM서브시스템(128k바이트까지 확장가)을 설치하였다.
- 256k 또는 512k바이트의 RAM서브시스템을 설치하였다.
- CMOS RAM(배터리 백업) 내장
- 리얼타임 클럭(배터리 백업) 내장
- 8개의 I/O 슬롯을 설치하였다.
  - (a) 6개의 32, 62핀 카드 에지코넥터
  - (b) 2개의 62핀 카드 에지코넥터

## AT의 하드웨어

PC AT의 블럭도를 그림 1에 나타낸다. 그림에서 보는바와 같이 CPU의 변경 및 인터럽트와 DMA채널이 증가한 것, 배터리 백업이 가능한 리얼타임 클럭이 탑재된 것 이외는 기본적으로는 PC XT와 거의 변함없다. 단, 키보드 콘트롤러를 시작으로 하여 어드레스 디코드나 여러가지의 제어신호의 작성에 ROM이나 PAL을 사용하고 있으며, 오픈 아키텍처라고 하는 것, 외견만으로는 알 수 없는 회로도 여기저기 있다. 그러나 이들은 클론방지라기 보다도 생산성향상을 위한 수단이라 생각하는 쪽이 좋을 것 같다. XT의 I/O슬롯의 8번째가 확장전용(외부박스용)으로 되어 있었지만 AT에서는 외부박스에 의한 확장은 할 수 없는 설계로 되어 버렸다.

시스템전체를 대충 전망하여 보면 LSI의 사용법 등이 순수하고, 세밀한 점에서는 각 IC의 기능을 헛되지 않게 구성하고 있다. 잘 숙련된 시스템이므로 80286시스템을 공부한

〔그림 1〕 IBM PC AT의 블록도

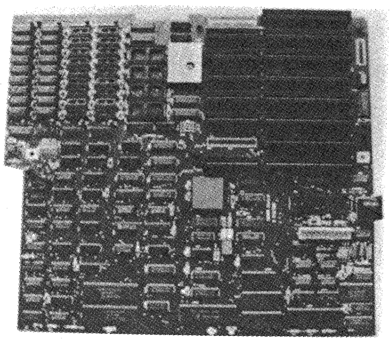



경우에는 좋은 교재가 되리라 생각한다. 단, 8비트시대의 오래된 LSI가 많이 사용되고 있는 것이 옥의 티라고 하나 할까?

## • AT의 메모리 맵

CPU가 8088에서 80286으로 바뀐 것에 의한 가장 큰 차이는 무엇보다도 메모리공간이 넓어진 것일 것이다. 그러나 기존의 소프트웨어를 그대로 시행시키지 않으면 안되기 때문에 그것 나름대로의 연구는 하고 있다. 메모리배치를 그림2에 나타낸다. MS-DOS/PC-DOS가 동작할 경우에는PC XT와 마찬가지로 메모리배치를 하지 않으면 안된다. 그런데 80286의 리세트 벡터는 리얼모드 시작은 아니지만 16M바이트공간의 FFFFF0H에 있으므로 전메모리공간의 최상위에 놓지 않으면 안된다. 따라서 BIOS-ROM의 어드레스 디코드를 조작하여 1M바이트, 16M바이트공간 모두 최상위에 위치하도록 연구하고 있다.

또 리얼모드의 MS-DOS 등에서는 1M바이트이상의 메모리는 관리되지 않으므로 RAM디스크 등으로서 사용한다.



〔사진 1〕 초기 PC AT의 마더 보드

AT의 최초의 버전(1984년 3월 발표)에서는 RAM에 128K비트의 것을 사용하여 패리티 체크용을 포함하여

128K비트×9비트×4뱅크=512K바이트

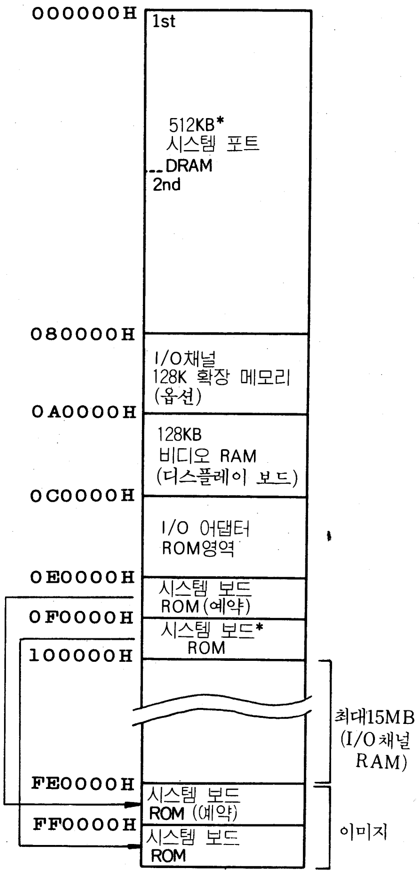
로 하였지만 최종 버전의 AT(1986년 3월 발표)에서는 256K비트의 RAM을 사용한다.

256K비트×9비트×2뱅크=512K바이트

로 시대배경에 민감하게 반응하고 있다.

ROM의 배치는 앞으로 설명한 바와같이 16비트버스로 되어있기 때문에 64K바이트의 공간에 16K바이트

〔그림 2〕 PC AT의 메모리 맵




*시스템 보드상에 실장된다.
기타는 I/O채널 옵션 또는 예약

ROM을 짝수, 홀수번지로 할당하지 않으면 안되기 때문에 32K바이트 ROM 을 2개 사용하고 있다.

이상이 시스템 보드에 표준으로 탑재되어 있는 모든 메모리이다. 비디오RAM은 디스플레이 어댑터에 탑재되어 있다.

## • AT의 I/O어드레스 맵

AT의 I/O어드레스공간은 1024바이트로 한정되어 있다. I/O어드레스 맵을 **그림3**에 나타낸다.

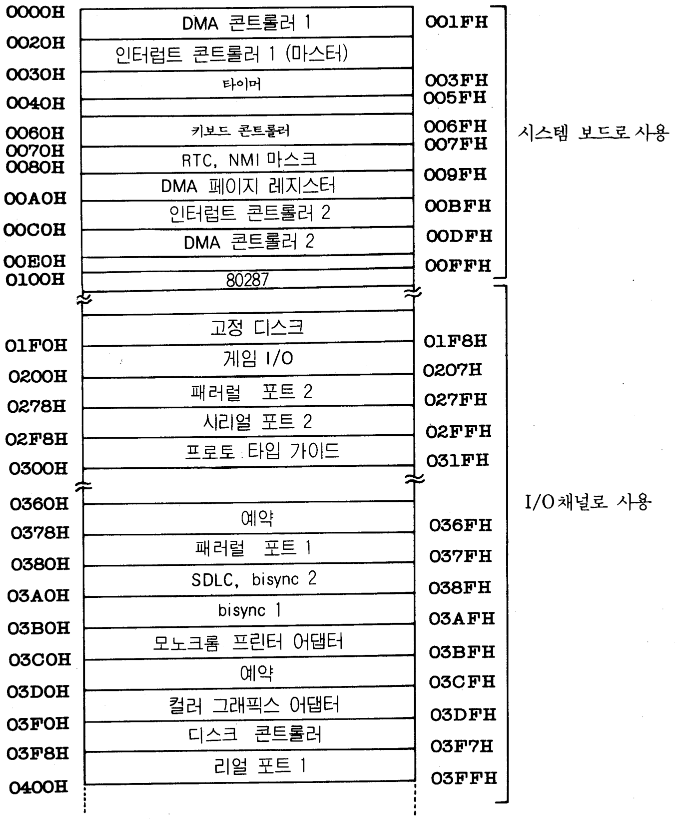
0000H~00FFH까지의 256바이트는 시스템보드 전용으로 되어 있어서 예약되어 있다. 또 나머지 0100H~03FFH도 확장보드로 사용하도록 되어 있어서 이들의 대부분이 옵션 모드를 위해서 예약되어 있다.

0000H~00FFH의 시스템보드에서 사용되는 I/O어드레스는 $A_3$~$A_9$를 74ALS139에서 디코드하여 칩셀렉트신호를 작성하고 있다. 따라서 시스템보드상의 주변디바이스는 32바이트단위로 I/O공간에 배열되어 있다. 시스템보드의 I/O디코드회로를 **그림4**에 나타낸다.

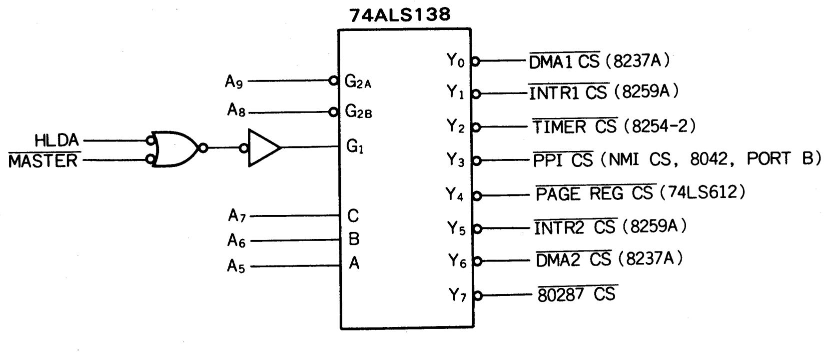
예외는 키보드 콘트롤러(8042)와 리얼타임 클럭(MC146818)이며 I/O어드레스 할당이 16바이트씩으로 되어 있다. 이 부분의 I/O어드레스 디코드는 256바이트ROM(74S288에 미리 프로그램되어 있어서 회로도만으로 내용은 정확하게 잡을 수는 없다.(**그림5, 그림6** 참고)

이 I/O공간은 키보드와 리얼타임 클럭의 기능만은 아니고 NMI마스크를 위한 게이트제어와 시스템에 필요한 기타의 여러가지 잡다한 I/O가 혼재하여 있다. 즉 AT의 시스템 보드내의 I/O로서 보드A, B의 2개가 BIOS의 POST(Power On Self Test)로 등록되어 있어서 쌍방 모두 8042의 속에 있는 것으로 되어 있다. 그러나 실은 그렇지는 않고, 보드B는 전혀 별개의 TTL래치(4비트)와

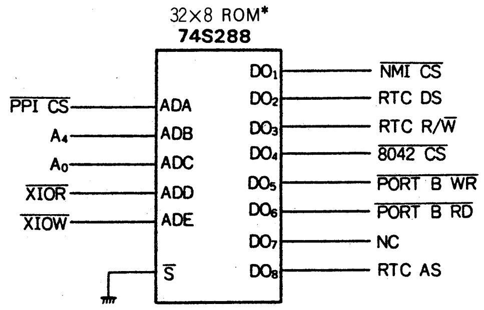
〔그림 3〕 PC AT의 I/O 맵




〔그림 4〕 시스템보드의 I/O 디코드회로




〔그림 5〕 포트 A, B의 디코드회로




*AT의 시스템 보드로부터 빼내 조정하면 이론이 성립하지 않는다.

〔그림 6〕 **PC-AT의 시스템보드상의 74S288의 내용**(어드레스 디코드 등으로부터 추측한 것)

| 어드레스 | | | | | HEX | 데이터 | | | | | | | | I/O 어드레스 선택 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADA | ADB | ADC | ADD | ADE | | $D_8$ | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | |
| 0 | 0 | 0 | 0 | 0 | 00 | 0 | × | 1 | 1 | 1 | × | 0 | 1 | —— |
| 0 | 0 | 0 | 0 | 1 | 01 | 0 | × | 1 | 1 | [0] | × | 0 | 1 | 60H, 64H |
| 0 | 0 | 0 | 1 | 0 | 02 | 0 | × | 1 | 1 | [0] | × | 0 | 1 | 60H, 64H |
| 0 | 0 | 0 | 1 | 1 | 03 | 0 | × | 1 | 1 | 1 | × | 0 | 1 | —— |
| 0 | 0 | 1 | 0 | 0 | 04 | 0 | × | 1 | 1 | 1 | × | 0 | 1 | —— |
| 0 | 0 | 1 | 0 | 1 | 05 | 0 | × | [0] | 1 | 1 | × | 0 | 1 | 61H |
| 0 | 0 | 1 | 1 | 0 | 06 | 0 | × | 1 | [0] | 1 | × | 0 | 1 | 61H |
| 0 | 0 | 1 | 1 | 1 | 07 | 0 | × | 1 | 1 | 1 | × | 0 | 1 | —— |
| 0 | 1 | 0 | 0 | 0 | 08 | 0 | × | 1 | 1 | 1 | × | 0 | 1 | —— |
| 0 | 1 | 0 | 0 | 1 | 09 | 0 | × | 1 | 1 | 1 | × | 0 | 1 | —— |
| 0 | 1 | 0 | 1 | 0 | 0A | [1] | × | 1 | 1 | 1 | [0] | 0 | [0] | 70H |
| 0 | 1 | 0 | 1 | 1 | 0B | 0 | × | 1 | 1 | 1 | × | 0 | 1 | —— |
| 0 | 1 | 1 | 0 | 0 | 0C | 0 | × | 1 | 1 | 1 | × | 0 | 1 | —— |
| 0 | 1 | 1 | 0 | 1 | 0D | 0 | × | 1 | 1 | 1 | [1] | [1] | 1 | 71H |
| 0 | 1 | 1 | 1 | 0 | 0E | 0 | × | 1 | 1 | 1 | [0] | [1] | 1 | 71H |
| 0 | 1 | 1 | 1 | 1 | 0F | 0 | × | 1 | 1 | 1 | × | 0 | 1 | —— |
| 1 | 0 | 0 | 0 | 0 | 10 | 0 | × | 1 | 1 | 1 | × | 0 | 1 | —— |
| 1 ~ 1 | 0 ~ 1 | 0 ~ 1 | 0 ~ 1 | 1 ~ 1 | 11 ~ 1F | 0 …… 0 | × …… × | 1 …… 1 | 1 …… 1 | 1 …… 1 | × …… × | 0 …… 0 | 1 …… 1 | —— |
| $\overline{\text{PPI CS}}$ | $A_4$ | $A_0$ | $\overline{\text{XIOR}}$ | $\overline{\text{XIOW}}$ | | RTC 어드레스 스트로브 | NC | $\overline{\text{포트 B리드}}$ | $\overline{\text{포트 B라이트}}$ | $\overline{\text{8042 CS}}$ | RTC R/$\overline{\text{W}}$ | RTC 데이터 스트로브 | $\overline{\text{NMI CS}}$ | |

3스테이트 버퍼(8비트)로 구성된 레지스터인 것이다(프로그래머로서 그 사양이 명확하면 회로구성은 아무래도 좋을 것이다).

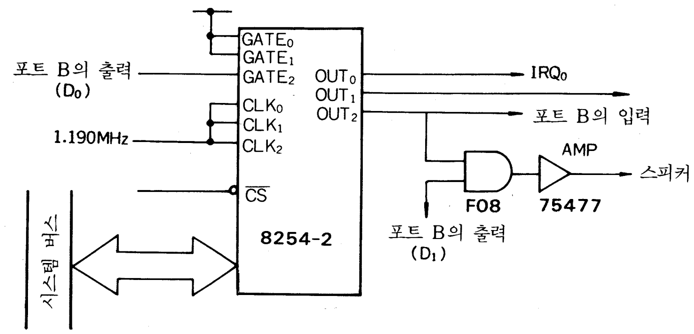
## • 시스템 타이머

시스템보드에는 시스템가동에 필요한 여러가지의 클럭을 발생시키는 프로그래머블 타이머를 내장하고 있다. 여기에는 8254-2프로그래머블 타이머 / 카운터가 사용되고 있다. 그 회로도를 그림7에 나타낸다.

8254-2에는 독립해서 프로그램 가능한 3개의 타이머 / 카운터가 있으며 채널0은 타이머 체크용의 인터럽트발생, 채널1은 다이나믹RAM을 위한 리플레시 클럭발생(15μs), 채널2는 스피커음원용으로 각각 용도가 결정되어 있다. 채널0은 PC XT에서 날짜 / 시간의 갱신에 사용되고 있었지만 PC AT에서는 리얼타임 클럭이 내장되었기 때문에 각종 I/O의 타임 아웃계측 등 각종의 용도에 이용할 수 있도록 되었다.

채널0, 1의 게이트는 항상"H"레벨로 풀업되어 있으므로 하드웨어적인 금지는 할 수 없다. 또 채널1은

〔그림 7〕 타이머회로



시스템상승시에 정해진 파라미터가 세트되므로 유저가 이것을 재프로그램하였을 경우 시스템의 동작은 보증되지 않는다(RAM의 내용이 소멸할 가능성이 있다). 채널2의 게이트는 I/O어드레스 61H의 비트0에 의해서 콘트롤할 수 있다. 또 각 타이머 / 카운터의 입력클럭은 1.190 MHz이다.

## • 시스템 인터럽트

PC AT의 80286의 하드웨어인터럽트는 그림8과 같이 1개의 NMI와 2개의 8259A를 사용한 16개의 IRQ에 의해서 구성되어 있다. NMI는 메모리 패리티 또는 I/O채널의 체크에 사용되고 있다. 한편 IRQ접수를 위한 2개의 8259A는 각각 콘트롤러1, 콘트롤러2라고 하며, 콘트롤러2의 INT출력은 콘트롤러1의 $IRQ_2$에 접속되어 있다.

우선순위는 물론 NMI가 가장 높고, 이어서 IRQ 0, 1, 2(8, 9, 10, 11, 12, 13, 14, 15), 3, 4, 5, 6, 7,의 순으로 되어 있다. 각 인터럽트기능을 표1에 나타낸다. 또 BIOS의 초기화루틴에서 인터럽트요구가 에지, 트리거로 프로그램되어 있다.

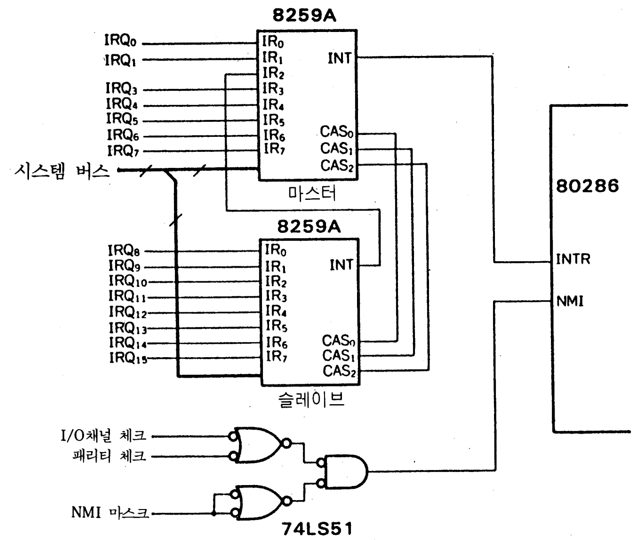
## • DMA콘트롤러

시스템보드는 2개의 DMA콘트롤러(8237A-5)를 IRQ와 마찬가지로 접속하여 7채널의 DMA를 지원하고 있다. 그림9에 DMA콘트롤러주변의 블럭도를 나타낸다.

좌측은 콘트롤러1이라 불리우며, I/O어댑터와 8 또는 16비트 시스템메모리와의 8비트 데이터의 전송에 사용하는 채널0~3까지 이며, 64K바이트 뱅크단위로 16M바이트의 어드레스공간을 액세스할 수 있다. BHE는 $A_0$를 반전한 것이 된다.

또 한쪽은 콘트롤러2라고 불리우며 채널4~7까지 있지만 채널4는 콘트롤러1에서 캐스케이트접속에 사

〔그림 8〕 인터럽트 콘트롤러




〔표 1〕 인터럽트 기능

| 인터럽트 | 인터럽트 요인 |
|---|---|
| NMI | 패리티 또는 I/O 채널 체크 |
| $IRQ_0$ | 타이머 출력 |
| $IRQ_1$ | 키보드 출력 버퍼 풀 |
| $IRQ_2$ | 콘트롤러 2로부터 캐스케이드 |
| $IRQ_3$ | 리얼포트 2 |
| $IRQ_4$ | 리얼포트 1 |
| $IRQ_5$ | 패러럴 포트 2 |
| $IRQ_6$ | 디스크 콘트롤러 |
| $IRQ_7$ | 패러럴 포트 1 |
| $IRQ_8$ | 리얼타임 클럭 |
| $IRQ_9$ | 소프트웨어 인터럽트(INT OAH) |
| $IRQ_{10}$ | 예 약 |
| $IRQ_{11}$ | 예 약 |
| $IRO_{12}$ | 예 약 |
| $IRQ_{13}$ | 80287 |
| $IRQ_{14}$ | 하드 디스크 |
| $IRQ_{15}$ | 예 약 |

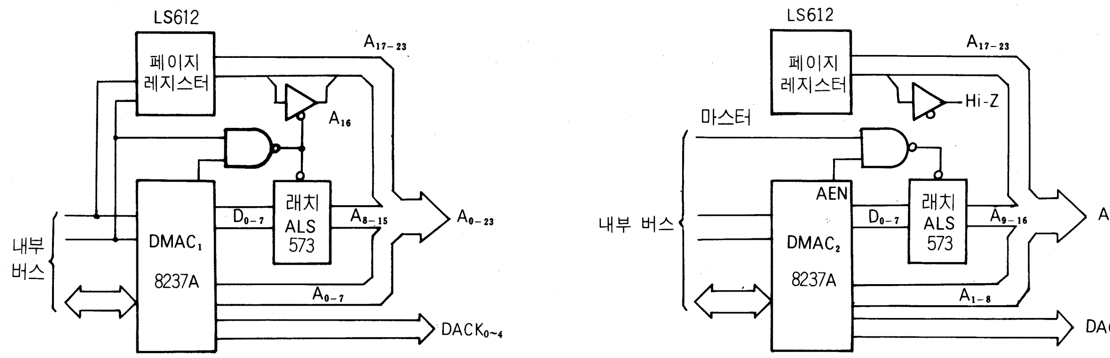
용되고 있다. 채널5~7은 16비트 데이터의 전송을 위해서 사용되며128K바이트 뱅크단위로 16M바이트 어드레스공간을 액세스할 수 있다. 즉 8237A에 접속되는 어드레스선을 상위방향으로 1비트 쉬프트하여 $A_1$~$A_{16}$으로 하고 있다. 이때 BHE와 $A_0$는 "L"레벨에 고정되어 있으므로 홀수바이트경계에서의 메모리조작은 할 수 없다. 표2에 각 채널의 용도를 나타낸다.

콘트롤러1 및 콘트롤러2는 각각 64K바이트경계를 128K바이트경계에 걸쳐서 전송할 수 없으므로 주의가 필요하다. 원래 8237A는 한번에 64M바이트의 전송밖에 할 수 없으므로 16K바이트의 어드레스공간을 액세스하기 위해서 상위어드레스선(채널0~3은 $A_{16}$~$A_{23}$, 채널5와 6은 $A_{17}$~$A_{23}$)을 출력하는 페이지 레지스터(74LS612)를 설치하고 있다.

XT에서는 단순한 래치였지만 AT에 와서 전용디바이스를 사용하여 기능을 올릴 수 있었다. 그러나 당시로서는 8237A를 사용하지 않을 수 없었던 것이지만 세련되지 않는 회로구성이다. 74LS612는 12비트×16의 RAM을 가지고 있어서 I/O버스에서 액세스하는 단순한 RAM으로서 동작하지만 DMA전송시에는 DMA 애크놀리지신호를 디코드한 메모리 어드레스의 내용이 상위어드레스 비트로서 출력된다.

표3에 페이지 레지스터의 I/O어드레스를 나타내었다. 단, 어드레스는 I/O공간상의 RAM으로서 사용

〔그림 9〕 DMA 콘트롤러




〔표 2〕 DMAC 각 채널의 용도

| 콘트롤러 | 채널 | 용　　도 |
|---|---|---|
| DMA1 | 0 | 예　비 |
| | 1 | SDLC |
| | 2 | 디스켓(IBM PC) |
| | 3 | 예　비 |
| DMA2 | 4 | 콘트롤러 1로 캐스케이드 |
| | 5 | 예　비 |
| | 6 | |
| | 7 | |

〔표 3〕 DMA 페이지 레지스터의 I/O 어드레스

| I/O어드레스 | 페이지 레지스터 |
|---|---|
| 080H | (＊1) |
| 081H | ch2 |
| 082H | ch3 |
| 083H | ch1 |
| 084H | |
| 085H | |
| 086H | |
| 087H | ch0 |
| 088H | |
| 089H | ch6 |
| 08AH | ch7 |
| 08BH | ch5 |
| 08CH | |
| 08DH | |
| 08EH | |
| 08FH | 리플레시 |

(＊1) 080H는 POST의 진단 프로그램으로 RAM으로서 사용

하지만 80H는 BIOS의 시스템진단 바이트로서 특정의 의미를 가지고 있다. 그러므로 유저가 이 어드레스의 내용을 바꾸어 쓰도록 하는 것은 하지 않는 편이 좋을 것이다.

AT에서는 이 디코드회로를 간단히 하기 위해 각 DMA페이지 레지스터의 어드레스 배치쪽을 분산시켜 놓고 있다. 심리적으로는 규칙적으로 나열하고 있는 것 이외의 프로그래밍을 할 때도 에러가 적어서 좋은 것이지만 AT의 하드웨어 엔지니어는 프로그래머가 이것을 커버하도록 하는 디자인으로 한 것 같다.

## • 키보드 콘트롤러

IBM PC시리즈의 키보드는 내부에 원칩 마이컴을 가진 말하자면 인텔리전트 키보드이다. 이 키보드와 시스템과의 데이터전송은 쌍방향의 시리얼통신으로 이루어진다. 그러나 모뎀을 사용하고 있는 것은 아니므로 전2중통신은 아니다. 코맨드에 의한 핸드쉐이크를 수반한 동기식 반2중통신이라는 것으로 된다. 따라서 키보드 케이블의 내용은 직류전원과 GND시리얼 데이터선, 클럭선의 4개만으로 되어 간단한 구성을 하고 있다.

시스템은 소프트웨어적으로 복잡한 AT키보드를 지원하지 않으면 안되지만 CPU가 이 처리를 하게 될 경우 본래의 처리수행에 영향을 미친다. 따라서 여기에도 키보드 콘트롤러로서 프로그래밍되었다. 8042에는 패러럴의 입출력포트가 있으며, 키보드와의 데이터통신만이 아니고 여기에 접속된 시스템관련의 중요한 I/O제어도 한다. 이 블럭도 를 그림10에 나타낸다.

## • 키보드 인터페이스

키보드 콘트롤러의 I/O어드레스 맵을 표4에, 데이터나 코맨드 플로우의 개념을 그림11에 나타낸다. 시스템을 스테이터스 레지스터의 내용을 보면서 입출력버퍼를 거쳐서 키보드와 데이터통신을 하고, 8042의 I/O포트의 제어나 여러가지의 진단프로그램의 실행은 시스템에서의 코맨드에 의해서 이루어진다. 이 때문에 키보드 콘트롤러는 단순한 시프트 레지스터나 통신포트는 아니고 그 역할은 요약하면

- 시스템에서 키보드에의 코맨드 송신
- 키보드에서 시스템에의 데이터(코맨드) 수신
- 송수신 데이터의 에러 체크
- 키 스캔코드의 변환
- 시스템에서의 코맨드 실행
- 송수신시의 키보드와의 핸드쉐이크

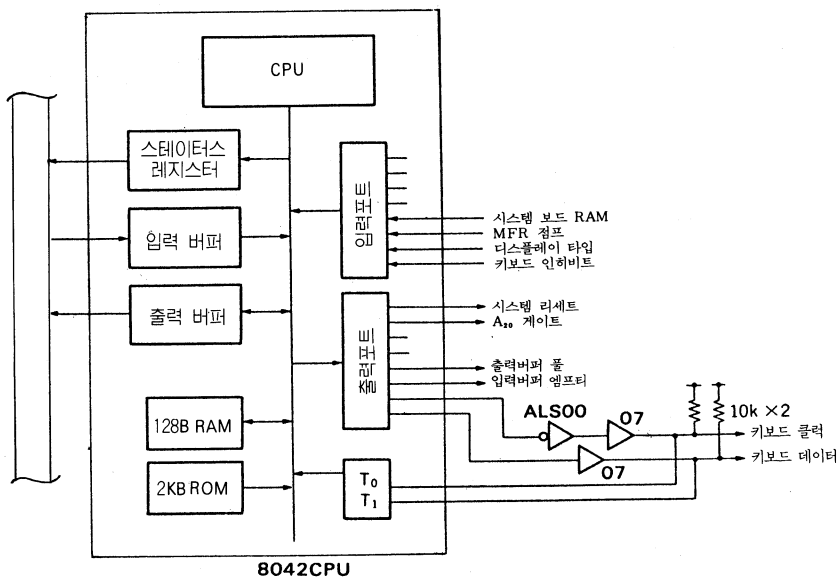
로 된다.

## • 키보드와의 통신

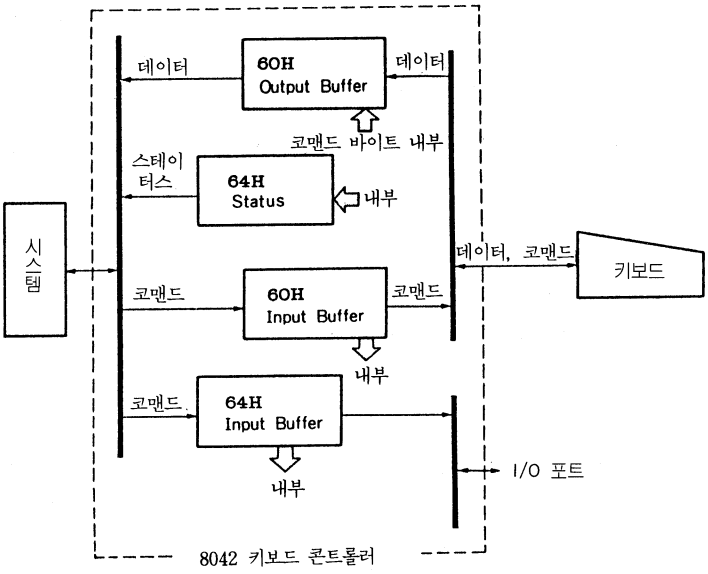
키보드와의 통신에는 키보드 데이터, 콘트롤러 데이터 모두 그림5에 나타낸 11비트 프레임의 시리얼 데이터가 사용된다. 또 동기클럭은 언제라도 키보드에서 출력된다.

콘트롤러는 키보드에서 데이터(

〔그림 10〕 보드 콘트롤러




〔그림 11〕 키보드 콘트롤러의 데이터 플로우 개념도




코맨드)를 접수하면 이것을 출력버퍼에 둠과 함께 시스템에 인터럽트를 건다. 이때 시스템이 그 데이터를 접수하기까지 콘트롤러는 인터페이스를 디스에이블로 하여 버린다. 그러나 이 수신처리에서 에러를 검지하면 스테이터스 레지스터에 에러 플래그를 세우므로 시스템은 그 내용을 알 수 있다.

수신시에 발생하는 에러는 패리티에러 또는 수신타임 아웃이다. 어느 경우도 키보드 데이터가 올바르게 접수되지 않는다고 하는 의미에서 콘트롤러의 출력버퍼에 FFH가, 스테이터스 레지스터에는 소정의 플래그가 세트된다. 또 패리티 에러일 때 콘트롤러는 재송요구를 내어 리트라이(Retry)를 시도한다.

〔표 4〕 키보드 콘트롤러의 I/O 어드레스

| 어드레스 | 리드 | 라이트 | 레지스터 선택 |
|---|---|---|---|
| 060H | 1 | 0 | 출력 버퍼 |
| | 0 | 1 | 입력 버퍼 |
| 064H | 1 | 0 | 스테이터스 레지스터 |
| | 0 | 1 | 코맨드 레지스터 |

〔표 5〕 시리얼 데이터 포맷(키보드)

| 비 트 | 내 용 |
|---|---|
| 1 | 스타트 비트 |
| 2 | 데이터 비트 0(LSB) |
| 3 | 데이터 비트 1 |
| 4 | 데이터 비트 2 |
| 5 | 데이터 비트 3 |
| 6 | 데이터 비트 4 |
| 7 | 데이터 비트 5 |
| 8 | 데이터 비트 6 |
| 9 | 데이터 비트 7(MSB) |
| 10 | 기수 패리티 |
| 11 | 스톱 비트 |

시스템에서 키보드에 데이터를 송신할 경우는 콘트롤러는 입력버퍼에 라이트된 코맨드데이터에 홀수패리티 비트를 삽입하여 시리얼로 변환하고 키보드에 송신한다. 키보드가 이 송신데이터를 정확하게 수신하였을 때에는 시스템에 액크놀리지를 되돌린다. 키보드에서의 응답이 없는 경우 시스템은 다음의 데이터를 키보드에 송신할 수는 없다.

키보드에의 송신에서 발생하는 에러는 송신타임 아웃, 패리티, 수신타임 아웃이 있으며, 이 경우 콘트롤러의 출력버퍼에 FFH가 스테이터스 레지스터에는 소정의 에러 플래그가 세트된다.

따라서 에러가 발생하였을 경우에는 출력버퍼의 내용이 FFH 또는 FEH 이므로 확인할 수 있으며, 그 원인은 스테이터스 레지스터를 참조하면 판명된다. 스테이터스 레지스터의 각 비트의 의미를 표6에 나타낸다.

〔표 6〕 스테이터스 레지스터의 각 비트의 의미

| 비트 | 스테이터스 | | 내용 |
|---|---|---|---|
| 0 | 출력버퍼 풀 | 0 | 출력 버퍼가 공백 |
| | | 1 | 출력 버퍼에 데이터가 있고 시스템이 리드되지 않는다. 시스템이 출력버퍼의 데이터를 리드하면 0로 된다. |
| 1 | 입력 버퍼 풀 | 0 | 입력버퍼가 공백 |
| | | 1 | 입력 버퍼에 라이트된 데이터가 남아 있다(콘트롤러가 리드되지 않는다). 콘트롤러가 리드하면 0로 된다. |
| 2 | 시스템 플래그 | 0 | 파워 온 리세트 후 |
| 3 | 코맨드/데이터 | 0 | 데이터 포트(060H)에 라이트 한다. |
| | | 1 | 코맨드 포트(064H)에 라이트한다. |
| 4 | 인히비트 스위치 | 0 | 키보드 인히비트 상태이다. |
| | | 1 | 키보드 인히비트 상태가 아니다. |
| 5 | 트랜스미트 타임 아웃 | 0 | 정상종료 |
| | | 1 | 키보드로의 전송이 불완전 |
| 6 | 리시브 타임 아웃 | 0 | 정상으로 수신 |
| | | 1 | 키보드로부터의 수신이 불완전 |
| 7 | 패리티 에러 | 0 | 기수 패리티로 수신(정상) |
| | | 1 | 우수 패리티로 수신(에러) |

키보드와의 통신은 핸드쉐이크에 의해서 쌍방향성이 유지되고 있는 것이지만 RS-232C와 같이 핸드쉐이크전용선이 있는 것은 아니고 PC 키보드 인터페이스에서는 데이터선, 클럭선의 레벨판정(쌍방 모두 오픈콜렉터로 드라이브되어 있다)에 의해서 실현되고 있다. 단, 이들의 제어권은 시스템측에 있다.

이 부근의 타이밍 등의 자세한 동작사양은 콘트롤러, 키보드 모두 프로그램된 원칩마이컴에 의해서 실현되고 있으므로 잘 알 수 없다. 요컨데 키보드에서의 데이터취득에 대해서는 BIOS에 그 서비스 루우틴이 있으므로 어지간한 것이 아닌 한 예를 들면 호환시스템을 연구하는 사람 이외는 이해할 필요는 없다고 하는 것이다. 단, 자신의 부담으로 키보드제어프로그램을 작성하는 사람을 위해서 각종 코맨드의 일람을 표7에 나타낸다.

## • 키보드의 스캔코드

키보드에서 전송되는 스캔코드는 콘트롤러 출력버퍼에 들어가기 전에 키보드 콘트롤러에 의해 변환을 받는다. 변환테이블을 표8에 나타낸다. 따라서 AT에서는 불완전하게 스캔된 키제어가 시스템에 보내지기 전에 시스템에 스캔코드로서 변환되고 더욱 BIOS가 이것을 ASCII코드로 변환한다고 하는 3단계의 처리가 이루어진다.

## • 키보드의 I/O포트

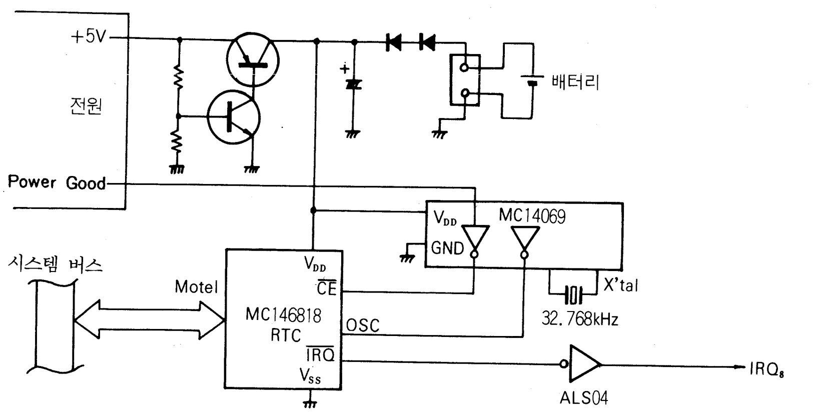
8042의 I/O포트는 앞에서 설명한 바와 같이 시스템관련의 중요한 기능을 담당하고 있다. 예를 들면 시스템리세트는 여기에서 발생하므로 프로텍트모드에서 리얼모드로 되돌리기 위해서는 이 I/O포트를 조작하지 않으면 안된다. 표9에 I/O 포트의 경우 각 비트 기능을 나타낸다(단, 이 I/O포트는 유저 혹은 응용 소프트웨어에서 조작하는 것은 아니고 모든 관리는 BIOS에 의해서 이루어지고 있다).

## • 리얼타임 클럭

PC XT에 없는 기능으로서 리얼타임 클럭이 있다. PC AT는 리얼타임 클럭과 64바이트의 배터리 백업가능한 CMOS RAM을 가진 모토롤러의 MC146818을 사용하고 있다. 이 주변회로를 그림 12에 나타낸다.

물론 클럭기능은 시간을 잘게 쪼개는 것이지만 CMOS RAM의 쪽은 시스템정보를 다음 회에 시스템이 부트되기까지 기억하여 두는 역할을 담당하고 있다. 표10 에 CMOS RAM의 각 바이트의 의미를 나타낸다. 배터리전원의 공급선은 MC146818과 CMOS인버터(MC14069)이다. PC AT의 전원유니트에서 출력되는 Power Good신호는 PC XT와 달

〔그림 12〕 리얼타임 클럭 주변회로



〔표 7〕 키보드 코맨드의 일람

| | |
|---|---|
| 20H | **Read Keyboard Controller's Command Byte**<br>60H로 전송된 키보드 콘트롤로의 코맨드를 출력버퍼에 라이트해 낸다. |
| 60H | **Write Keyboard Controller's Command Byte**<br>I/O 어드레스 60H에 키보드 콘트롤러로의 코맨드를 라이트.<br>7 6 5 4 3 2 1 0<br>0: "1"로 출력버퍼 풀 인터럽트를 인에이블로 한다.<br>1: "0" 예약<br>2: 이 비트는 스테이터스 레지스터의 시스템 플래그에 반영한다.<br>3: "1"에서 키보드 금지를 디스어셈블한다.<br>4: "1"에서 키보드 금지상태로 한다.<br>5: "1"에서 IBM PC 모드로 한다(패리티 체크 또는 스캔 코드 변환을 하지 않는다).<br>6: "1"에서 IBM PC 호환모드로 한다(2바이트 브레이크 시퀀스를 1바이트 IBM PC 포맷으로 변환).<br>7: "0" (예약) |
| AAH | **Self Test**<br>키보드 콘트롤러의 샘플 테스트. 출력버퍼에 55H가 출력되면 정상 종료. |
| ABH | **Interface Test**<br>키보드의 클럭과 데이터선의 체크. 출력버퍼에 55H가 출력되면 정상 종료.<br>00H : 에러없음<br>01H : 키보드 클럭선이 "L"레벨<br>02H : 키보드 클럭선이 "H"레벨<br>03H : 키보드 데이터선이 "L"레벨<br>04H : 키보드 데이터선이 "H"레벨 |
| ACH | **Diagnostic Dump**<br>콘트롤러내의 정보를 시스템에 스캔 코드 포맷으로 덤프한다.<br>내용 : 16B RAM, 입출력 포트, 스테이터스 워드 |
| ADH | **Disable Keyboard Interface**<br>키보드 클럭선을 "L"레벨로 하고, 키보드와의 데이터 전송을 디스어셈블한다. 결과적으로 콘트롤러 코맨드의 비트 4를 세트한다. |
| AEH | **Enable Keyboard Interface**<br>키보드 인터페이스를 재개한다. 콘트롤러 코맨드의 비트 4는 클리어된다. |
| COH | **Read Input Port**<br>출력버퍼에 입력포트의 내용을 출력한다. 본 코맨드는 출력버퍼가 비어 있을 때만 유효. |
| DOH | **Read Output Port**<br>출력포트의 내용을 출력버퍼에 출력한다. 본 코맨드는 버퍼가 비어 있을 때만 유효. |
| D1H | **Write Output Port**<br>I/O 어드레스 60H에 라이트된 다음 바이트 데이터가 출력버퍼에 출력된다.<br>주의 : 비트 "0"에 "0"를 라이트하면 시스템이 리세트 된다. |
| EOH | **Read Test Input**<br>$T_0$, $T_1$입력을 출력버퍼의 비트 0, 1에 출력한다. |
| FO~FFH | **Pulse Output Port**<br>코맨드의 하위 4비트중 몇개가 "L"레벨이면 대응하는 출력포트에 약 6$\mu$s간 "L"레벨이 출력된다. 시스템 리세트는 이 코맨드로 행한다. |

라서 MC14069에서 반전하여 MC146818 의 CE에 들어 간다. 그러므로 전원라인의 공급이 끊어지면 MC146818 의 입출력핀이 디스에이블로 되어 소비전력을 최소한으로 억제한다. 또 여기에 사용되고 있는 배터리회로는 다이오드만의 재충전이 불가능한간단한 것이므로 방전하여 버리면 시스템정보는 소멸한다.

MC146818은 어느 형태의 버스에도 간단히 접속가능한 인터페이스를 가지고 있다. 따서 AT버스에도 쉽게 접속이 가능하다. 그러나 클럭용의 레지스터나 CMOS RAM에의 액세스는 I/O공간의 2개의 어드레스(70H, 71H)를 사용하여 MC146818의 내부어드레스 지정, 데이터의 입출력과 2회의 조작이 필요하다.

내부어드레스 00H~0DH는 클럭이나 스테이터스 레지스터로서 MC146818로 처음부터 결정되어 있으므로 상세한 것은 MC146818의 데이터북을 찾아봐야 한다.

## AT 버스

IBM PC관계에서 가장 흥미가 있는 내용은 역시 I/O채널일 것이다.

I/O채널 카드에지 코넥터의 물리적인 기판상의 배치를 **그림 13**에 나타낸다. 앞에서 설명한 바와 같이 AT로 되어 24비트의 어드레스공간을 지원하지 않으면 안되었던 형편상 XT까지의 62핀의 I/O슬롯에서는 쓸 수 없게 되어 6개의 36핀 I/O 슬롯을 첨가하고 있다. 각 I/O 채널의 핀번호와 신호의 내용(기능)을 **그림14**와 **표11** 에 나타낸다.

I/O채널의 모든 신호선은 TTL 호환이지만 확장보드측 입력은 최대 2개의LS-TTL부하까지 드라이브 가능하다. 즉 확장보드의 입력은 반드시 버퍼링영역이라는 것이다. 실제 시스템보드의 I/O채널출력은 LS 또는 ALS-TTL 버스버퍼에 의해서 드라이브되고 있다. 각 신호의 시스템측 입출력디바이스를 **그림 15**에 나타낸다.

### • I/O채널 코넥터신호

다음에 I/O채널 코넥터에 나오고 있는 신호에 대해서 상세하게 설명한다. **그림11**을 보면서 확인하기 바란다.

**• $SA_0$~$SA_{19}$(쌍방향)**

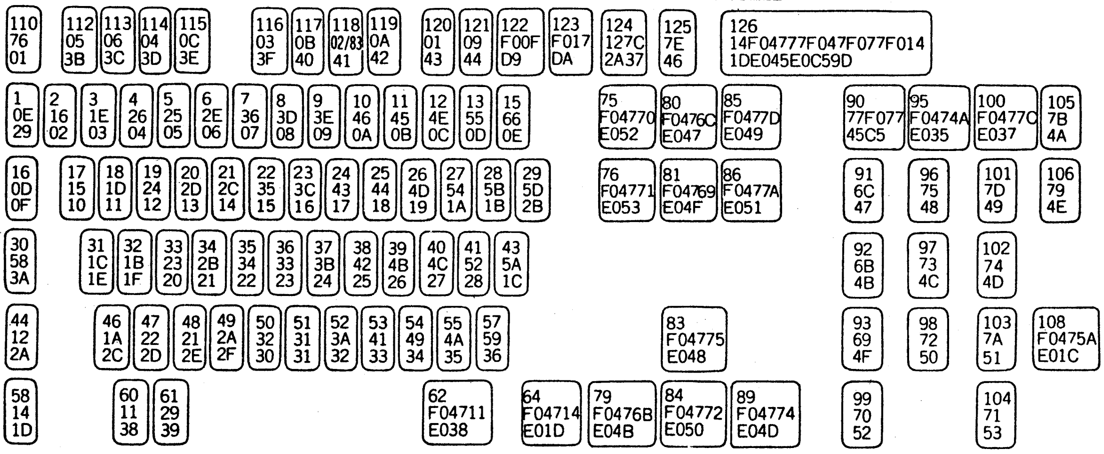
하위20비트의 어드레스버스(CPU 또는 DMA콘트롤러에서 출력된다)이다. XT의 8088 IM바이트 어드르레스와 호환성을 유지하기 위해서 24개의 어드레스선을 분할하고 있다. $SA_0$~$SA_{19}$는 BALE신호에 의해 배치된다. 이들이 쌍방향성인것은 다른 CPU나 DMA 콘트롤러가 이 버

〔표 8〕 스캔코드의 변환 테이블(101/102 키보드 배열)

▶아래의 배치는 키보드의 배치에 준한다.
▶숫자는 위에서부터 순서대로

Key Number
Keyboard Scan Code
System Scan Code

를 의미하고 있다.




〔표 9〕 키보드 콘트롤러의 I/O 포트기능

●입력 포트

| 비트 | 내용 |
|---|---|
| 0 | 미사용 |
| 1 | 미사용 |
| 2 | 미사용 |
| 3 | 미사용 |
| 4 | 시스템 보드 메모리("0"=2nd 256kB RAM 디스어셈블, "1"=인에이블 |
| 5 | 매뉴 픽처 점프("0"=점프결과, "1"=점프안됨) |
| 6 | 디스플레이 타입("0"=프라이머리가 CGA, "1"=프라이머리가 MDA) |
| 7 | 키보드 인히비트("0"=인히비트, "1"=인히비트 해제) |

●출력 포트

| 비 트 | 내 용 |
|---|---|
| 0 | 시스템 리세트 |
| 1 | 게이트 $A_{20}$ |
| 2 | 미사용 |
| 3 | 미사용 |
| | 출력 버퍼 풀 |
| 5 | 입력 버퍼 엠피티 |
| 6 | 키보드 클럭(출력) |
| 7 | 키보드 데이터(출력) |

●테스트 입력포트

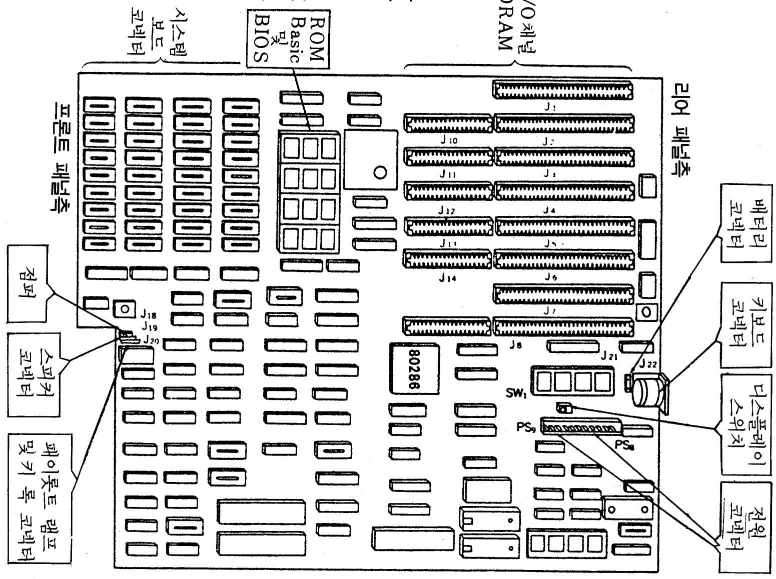
| | |
|---|---|
| $T_0$ | 키보드 클럭(입력) |
| $T_1$ | 키보드 데이터(입력) |

스를 드라이브할 수 있도록 하기 위해서이다. 즉 장래의 멀티프로세서 시스템에 대응할 수 있도록 설계되어 있는 것이다.

● $LA_{17}$~$LA_{23}$(쌍방향)

상위7비트의 어드레스선인데 SA와의 차이는 래치되지 않은 채로 I/O채널에 나오고 있는 것이며, BALE가"H"레벨일 때 확정된다. 래치는

〔그림 13〕 IBM PC AT의 시스템 보드 배치도



〔표 10〕 CMOS RAM의 각 바이트의 의미

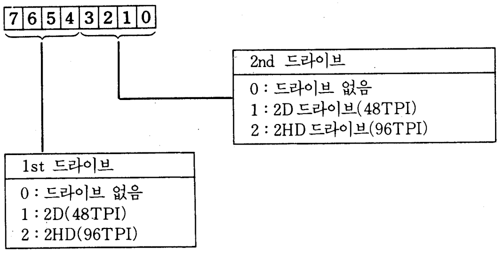
●진단정보(OE)

| 비 트 | 내 용 |
|---|---|
| 7 | 0 : RTC 배터리 정상<br>1 : RTC 배터리 불량 |
| 66<br>6 | 0 : 체크섬 정상<br>1 : 체크섬 에러 |
| 5 | 0 : 구성정보 정상<br>1 : 구성정보 불확정(디스크 드라이브가 없다. 디스플레이 설정과 어댑터 종류의 불일치) |
| 4 | 0 : 메모리 사이즈 정상<br>1 : 메모리 사이즈 불일치(구성정보와 파워 ON 진단과의 비교) |
| 3 | 0 : 하드 디스크 이니셜라이즈(하드 디스크로부터 부트 가능)<br>1 : 하드 디스크 이니셜라이즈에 실패(부트 불가능) |
| 2 | 0 : 시스템시각 확정<br>1 : 시스템 불확정 |
| 1<br>0 | } 예약 |

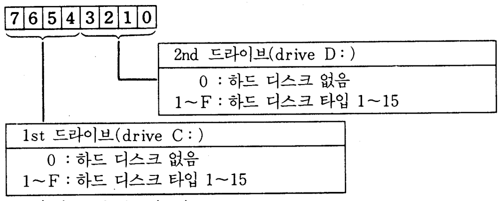
●쇼트다운 스테이터스(OF)

| Hex | 내 용 |
|---|---|
| 00H | 보통 리세트 시퀀스 |
| 01H | 메모리 사이즈 체크후의 쇼트 다운 |
| 02H | 메모리 테스트 후의 쇼트 다운 |
| 03H | 메모리 에러시의 쇼트 다운 |
| 04H | 부트 로더 요구의 쇼트 다운 |
| 05H | 옵션 I/O ROM의 초기화 (타이머 인터럽트 있음) |
| 06H | 프로텍트 모드 테스트 7통과 |
| 07H | 프로텍트 모드 테스트 7실패 |
| 08H | 프로텍트 모드 테스트 1실패 |
| 09H | 블록 이동시의 쇼트 다운 요구 |
| 0AH | 옵션 I/O ROM 초기화 (타이머 인터럽트 없음) |

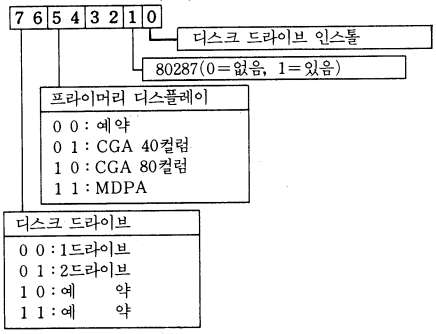
●디스크 드라이브 타입(10)




●하드 디스크 타입(12)




●기기구성정보(14)




●베이스 메모리 사이즈(15, 16)

Hex16 Hex15

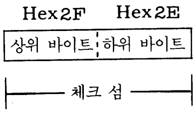
| 상위 바이트 | 하위 바이트 | |
|---|---|---|
| 01 | 00 | 256kB 시스템 보드 RAM |
| 02 | 00 | 512kB 시스템 보드 RAM |
| 02 | 80 | 640kB RAM (512k 시스템+128k I/O 확장 메모리) |

●확장 메모리 사이즈(17, 18)

Hex18 Hex17

| 상위 바이트 | 하위 바이트 | |
|---|---|---|
| 02 | 00 | 512kB I/O 어댑터 1매 |
| 04 | 00 | 512kB I/O 어댑터 2매 |
| 06 | 00 | 512kB I/O 어댑터 3매 |
| ⋮ | ⋮ | |
| 3C | 00 | 최대 15MB I/O 어댑터 |

●체크 섬(2E, 2F)




●확장 메모리 사이즈(30, 31)

Hex31 Hex30

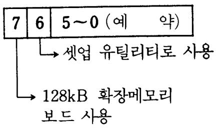
| 상위 바이트 | 하위 바이트 |
|---|---|

(내용은 17, 18과 마찬가지로 파워 ON시에 체크된 1MB 이상의 메모리 공간의 실장 메모리 용량. INT 15H에서 시스템에 통지된다.)

●서기 상위 2자리의 내용(32)

| 상위 BCD | 하위 BCD |
|---|---|

●인포메이션 플래그(33)




확장보드측에서 이루어지지 않으면 안된다. SA와 LA의 조합에 의해서 비로소 16M바이트의 어드레스공간이 얻어진다.

LA도 쌍방향성이다.

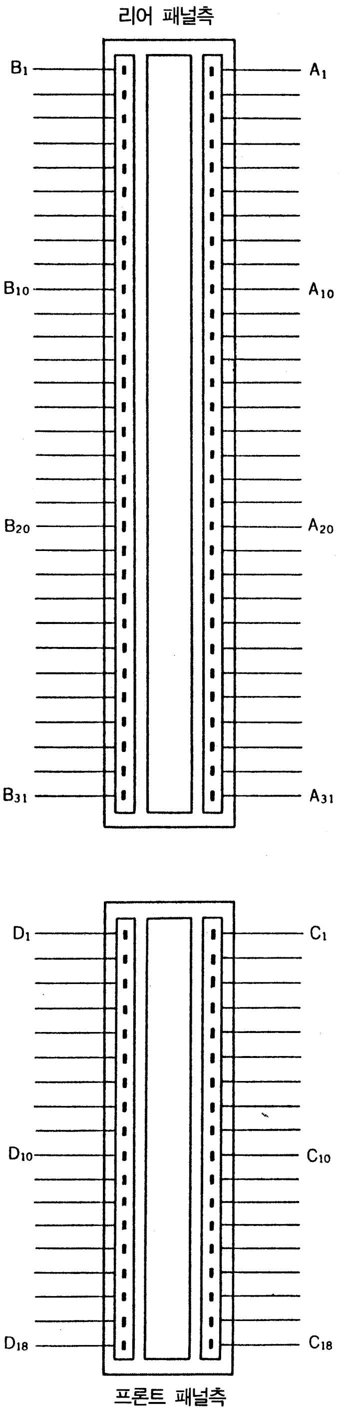
● CLK(출력)

6MHz 또는 8MHz의 시스템클럭이다(호환기에 따라서는 동작 클럭 주파수가 다르므로 주의하기 바란다).

● RESET DRV(출력)

이른바 리세트신호이다. AT에서는 전원 ON시 또는 AC전원이 낮을 때"H"레벨로 된다. 이것은 출력신호이므로 하드웨어 리세트기능은 없다. 시스템보드의 스테이터스와 확

〔그림 14〕 I/O채널 코넥터




장I/O의 리세트시퀀스에 사용된다.

• $SD_0$~$SD_{15}$ (쌍방향)

16비트 데이터버스이며, $SD_0$가 LSB, $SD_{15}$가 MSB이다. I/O디바이스의 바이트 데이터의 전송에는

〔표 11〕 I/O채널의 핀번호

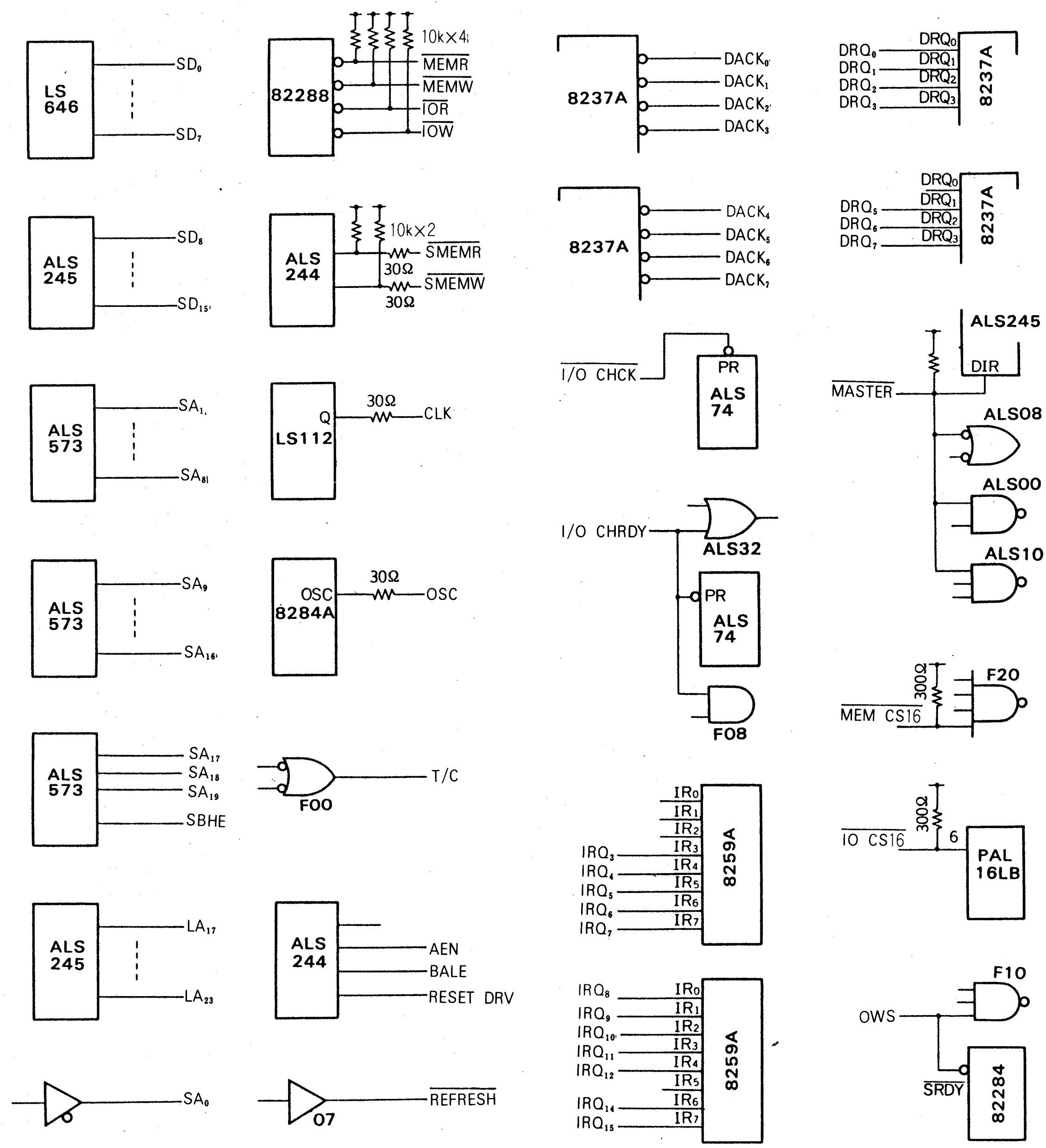
| 핀번호 | 신호명 | I/O |
|---|---|---|
| $B_1$ | GND | 그라운드 |
| $B_2$ | RESET DRV | O |
| $B_3$ | $+5V_{DC}$ | 전원 |
| $B_4$ | $IRQ_9$ | I |
| $B_5$ | $-5V_{DC}$ | 전원 |
| $B_6$ | $DRQ_2$ | I |
| $B_7$ | $-12V_{DC}$ | 전원 |
| $B_8$ | 0 WS | I |
| $B_9$ | $+12V_{DC}$ | 전원 |
| $B_{10}$ | GND | 그라운드 |
| $B_{11}$ | −SMEMW | O |
| $B_{12}$ | −SMEMR | O |
| $B_{13}$ | −IOW | I/O |
| $B_{14}$ | −IOR | I/O |
| $B_{15}$ | $-DACK_3$ | O |
| $B_{16}$ | $DRQ_3$ | I |
| $B_{17}$ | $-DACK_1$ | O |
| $B_{18}$ | $DRQ_1$ | I |
| $B_{19}$ | −Refresh | I/O |
| $B_{20}$ | CLK | O |
| $B_{21}$ | $IRQ_7$ | I |
| $B_{22}$ | $IRQ_6$ | I |
| $B_{23}$ | $IRQ_5$ | I |
| $B_{24}$ | $IRQ_4$ | I |
| $B_{25}$ | $IRQ_3$ | I |
| $B_{26}$ | $-DACK_2$ | O |
| $B_{27}$ | T/C | O |
| $B_{28}$ | BALE | O |
| $B_{29}$ | $+5V_{DC}$ | 전원 |
| $B_{30}$ | OSC | O |
| $B_{31}$ | GND | 그라운드 |

| 핀번호 | 신호명 | I/O |
|---|---|---|
| $A_1$ | −I/O CH CK | I |
| $A_2$ | $SD_7$ | I/O |
| $A_3$ | $SD_6$ | I/O |
| $A_4$ | $SD_5$ | I/O |
| $A_5$ | $SD_4$ | I/O |
| $A_6$ | $SD_3$ | I/O |
| $A_7$ | $SD_2$ | I/O |
| $A_8$ | $SD_1$ | I/O |
| $A_9$ | $SD_0$ | I/O |
| $A_{10}$ | I/O CH RDY | I |
| $A_{11}$ | AEN | O |
| $A_{12}$ | $SA_{19}$ | I/O |
| $A_{13}$ | $SA_{18}$ | I/O |
| $A_{14}$ | $SA_{17}$ | I/O |
| $A_{15}$ | $SA_{16}$ | I/O |
| $A_{16}$ | $SA_{15}$ | I/O |
| $A_{17}$ | $SA_{14}$ | I/O |
| $A_{18}$ | $SA_{13}$ | I/O |
| $A_{19}$ | $SA_{12}$ | I/O |
| $A_{20}$ | $SA_{11}$ | I/O |
| $A_{21}$ | $SA_{10}$ | I/O |
| $A_{22}$ | $SA_9$ | I/O |
| $A_{23}$ | $SA_8$ | I/O |
| $A_{24}$ | $SA_7$ | I/O |
| $A_{25}$ | $SA_6$ | I/O |
| $A_{26}$ | $SA_5$ | I/O |
| $A_{27}$ | $SA_4$ | I/O |
| $A_{28}$ | $SA_3$ | I/O |
| $A_{29}$ | $SA_2$ | I/O |
| $A_{30}$ | $SA_1$ | I/O |
| $A_{31}$ | $SA_0$ | I/O |

| 핀번호 | 신호명 | I/O |
|---|---|---|
| $D_1$ | −MEM CS16 | I |
| $D_2$ | −I/O CS16 | I |
| $D_3$ | $IRQ_{10}$ | I |
| $D_4$ | $IRQ_{11}$ | I |
| $D_5$ | $IRQ_{12}$ | I |
| $D_6$ | $IRQ_{15}$ | I |
| $D_7$ | $IRQ_{14}$ | I |
| $D_8$ | $-DACK_0$ | O |
| $D_9$ | $DRQ_0$ | I |
| $D_{10}$ | $-DACK_5$ | O |
| $D_{11}$ | $DRQ_5$ | I |
| $D_{12}$ | $-DACK_6$ | O |
| $D_{13}$ | $DRQ_6$ | I |
| $D_{14}$ | $-DACK_7$ | O |
| $D_{15}$ | $DRQ_7$ | I |
| $D_{16}$ | $+5\ V_{DC}$ | 전원 |
| $D_{17}$ | −MASTER | I |
| $D_{18}$ | GND | 그라운드 |

| 핀번호 | 신호명 | I/O |
|---|---|---|
| $C_1$ | SBHE | I/O |
| $C_2$ | $LA_{23}$ | I/O |
| $C_3$ | $LA_{22}$ | I/O |
| $C_4$ | $LA_{21}$ | I/O |
| $C_5$ | $LA_{20}$ | I/O |
| $C_6$ | $LA_{19}$ | I/O |
| $C_7$ | $LA_{18}$ | I/O |
| $C_8$ | $LA_{17}$ | I/O |
| $C_9$ | −MEMR | I/O |
| $C_{10}$ | −MEMW | I/O |
| $C_{11}$ | $SD_{08}$ | I/O |
| $C_{12}$ | $SD_{09}$ | I/O |
| $C_{13}$ | $SD_{10}$ | I/O |
| $C_{14}$ | $SD_{11}$ | I/O |
| $C_{15}$ | $SD_{12}$ | I/O |
| $C_{16}$ | $SD_{13}$ | I/O |
| $C_{17}$ | $SD_{14}$ | I/O |
| $C_{18}$ | $SD_{15}$ | I/O |

〔그림 15〕 I/O채널 입출력 디바이스




하위8비트($SD_{0\sim7}$)를, 워드데이터의 전송에서는 8비트 디바이스의 경우 2회에 나누어 바이트 데이터를, 16비트 디바이스에서는 $SD_{0\sim15}$를 사용하여 1회에 전송을 한다.

• BALE(버퍼출력)

82288버스 콘트롤러가 출력하는 어드레스 래치 인에이블신호의 버퍼링출력이다. CPU 또는 DMA콘트롤러의 어드레스확정을 나타내기 위한 신호이다. $SA_{0-19}$는 여기서 래치되고 있다. 또 DMA사이클중은 "H"레벨로 유지되고 있다.

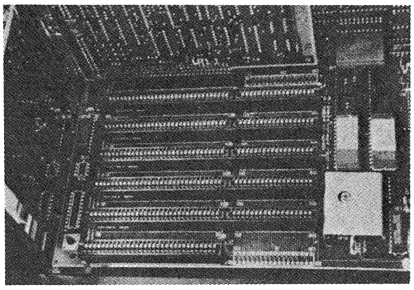
• I/O CH CK(입력)

메모리 또는 I/O채널상의 디바이스의 패리티 에러정보이다. 따라서 이 신호가 액티브일 때에는 수정불가능한 시스템 에러라는 것으로 된다.

• I/O CH RDY(입력)

이 신호는 메모리나 I/O디바이스가 약간 긴듯한 메모리 또는 I/O 사이클을 필요로할 때에 사용된다. 이와 같은 늦은 디바이스는 어드레스가 확정되었을 때 또는 라이트, 리드의 지시를 받았을 때에 곧"L"레벨로 하지 않으면 안된다. 머신사이클은 클럭 사이클(167ns)의 정수배로 늘이는 것이 가능하지만 2.5ms이상 "L"레벨로 해서는 안된다.

〔사진 2〕 PC AT의 I/O 슬롯의 형태



• $IRQ_{3-7}$, $IRQ_{9-12}$, $IRQ_{14,15}$ (입력)

CPU에 대한 I/O디바이스에서의 하드웨어 인터럽트신호이다. 우선순위는 $IRQ_9$가 가장 높고, 이하 10, 11, 12, 14, 15, 3, 4, 5, 6, 7의 순으로 낮아진다. 인터럽트는 "L"에서 "H"로의 상승이 발생하여 CPU가 확인하기까지 "H"레벨을 유지하지 않으면 안된다. 또 $IRQ_{13}$은 시스템보드이며, $IRQ_8$은 리얼타임 클럭으로 각각 사용되고 있으므로 I/O채널상의 확장보드에서 이용할 수 있다.

• IOR (쌍방향)

I/O디바이스에 대해서의 리드명령이다. 물론 CPU 또는 DMA콘트롤러에서 나오는 신호이지만 I/O채널에 있는 별도의 CPU나 DMA 콘트롤러에서 나오는 것도 있으므로 쌍방향으로 된다. 부논리신호이다.

• IOW (쌍방향)

I/O디바이스에 데이터버스의 데이터를 라이트시키는 신호이다. I/O디바이스에 데이터 버스의 데이터를 읽어 들이게 되어 있다. CPU 또는 DMA콘트롤러에서 나오는 신호이다. 부논리신호이다.

• SMEMR (출력), −MEMR (쌍방향)

메모리의 리드신호이다(메모리→데이터버스). −SMEMR은 1M바이트공간(000000H~0FFFFFH) 전용의 리드신호이다. −MEMR은 모든 메모리 사이클에서 액티브로 된다. 이 신호는 쌍방향이므로 I/O채널에서도 드라이브할 수 있지만 −MEMR은 액티브로 하기 전에 1클럭 사이클의 기간, 버스상의 어드레스가 확정되어 있지 않으면 안된다. −S-MEMR, −MEMR 모두 부논리이다.

• −SMEMW (출력), −MEMW (쌍방향)

이 신호는 메모리에 라이트를 지시하는 점을 제외하고 −SMEMR, −MEMR와 같다.

• $DRQ_0$~$DRQ_3$, $DRQ_5$~$DRQ_7$ (입력)

I/O디바이스 또는 I/O채널상의 CPU에서 시스템에 대해서의 DMA 요구신호선이다. 우선순위는 $DRQ_0$가 가장 높고, $DRQ_7$이 가장 낮아지고 있다. DMA요구는 "H"레벨에서 발생하고, DACK(DMA요구액크놀리지)가·액티브로 되기까지 그 레벨을 유지하고 있지 않으면 안된다. $DRQ_{0}$~$_{3}$은 8비트전송에, $DRQ_5$~$_7$은 16비트전송에 사용된다. 또 $DRQ_4$는 시스템보드에서 사용되기 때문에 I/Q채널에서는 이용할 수 없다.

• $DACK_0$~−$DACK_3$, −$DACK_5$~−$DACK_7$ (출력)

앞에서 설명한 $DRQ_X$에 대한 부논리의 액크놀리지신호이다.

• ACN (출력)

이 신호가 액티브 되었을 때 DMA 콘트롤러는 어드레스 버스 및 데이터 버스의 리드 / 라이트를 콘트롤한다.

• −REFRESH (쌍방향)

이 신호는 리플레쉬 사이클을 나타내기 위해서 사용된다. 또 I/O채널상에서 CPU에 의해서 드라이브되는 것이 가능하다.

• T/C (출력)

DMA전송이 종료된 것을 나타내는 신호이다.

• SBHE (쌍방향)

데이터 버스의 상위바이트($SD_8$~$SD_{15}$) 전송을 나타내는 신호이다. 하위바이트전송은 $SA_0$에서 검출할 수 있다. 어댑터 포트내의 데이터버퍼의 게이트신호로서 사용할 수 있다.

• −MASIER (입력)

DRQ신호와 함께 시스템 제어를 얻기 위해서 사용된다. I/O채널상의 CPU 또는 DMA콘트롤러는 캐스케이드 모드에서 DRQ를 내고, −DACK 를 얻는다. −DACK를 접수하면 I/O채널의 CPU는 −MASER를 "L"레벨로 하여 어드레스, 데이터 및 콘트롤 버스의 제어를 허락한다.

−MASER가 "L"레벨로 유지되어도 I/O채널의 CPU는 1클럭 사이클의 사이 어드레스 및 데이터 버스의 드라이브를 기다리지 않으면 안된다. 또 2클럭 사이클의 사이 리드/라이트 신호를 낼 수 없다. 15μs이상 "L"레벨로 유지하면 리플레쉬 불능기간이 발생하여 메모리상의 데이터가 소멸할 가능성이 있다.

• −MEM CS16 (입력)

주어진 데이터전송이 1웨이트 스테이트의 16비트 메모리 사이클이면 시스템보드에 대해서 이 신호를 출력하지 않으면 안된다. 또 이것은 $LA_{17}$~$LA_{23}$을 디코드하여 만들어 내지 않으면 안된다. 이 신호는 오픈콜렉터 또는 20mA의 싱크능력이 있는 3 스테이트 버퍼에서 드라이브하지 않으면 안된다.

• −I/O CS16 (입력)

주어진 데이터전송이 1웨이트 스테이트의 16비트 I/O사이클이면 시스템 보드에 대해서 이 신호를 출력하지 않으면 안된다. 또 이것은 어드레스선을 디코드하며 만들어 내지 않으면 안된다.

• OSC (출력)

14.31818MHz(70μs) 클럭이다. 시스템클럭과는 동기하고 있지 않다. 2분주로 컬러 버스트주파수가 얻어진다.

• OWS (입력)

이것은 제로 웨이트스테이트라고 읽는다. 즉 웨이트사이클이 전혀 없는 버스사이클인 것을 CPU에 알

려주는 신호이다. 이 신호는 오픈콜렉터 또는 20mA의 싱크능력이 있는 3 스테이트버퍼에서 드라이브하지 않으면 안된다.

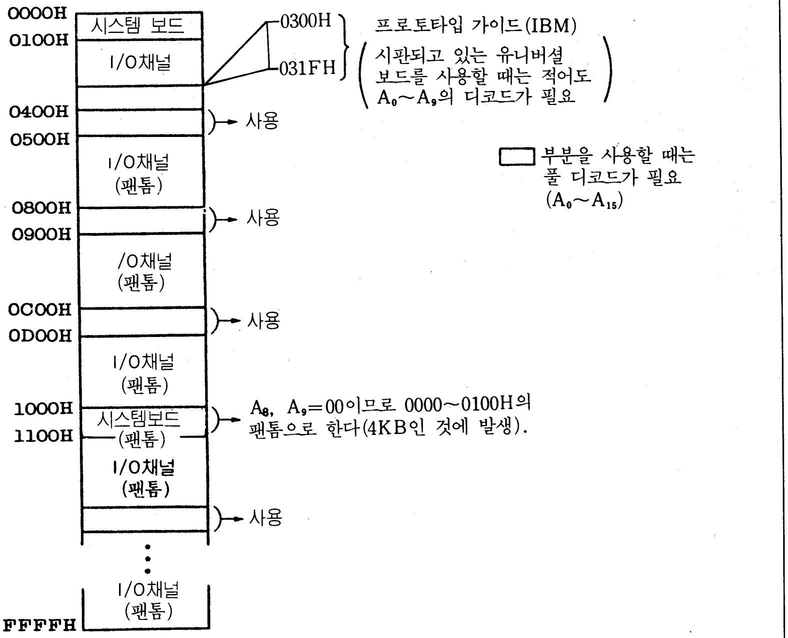
## • I/O 어댑터보드

I/O어드레스 맵에는 IBM의 프로토타입 카드(자작용의 유니버셜 기판)를 위한 영역이 확보되어 있다. 그러나 300H~31FH의 32바이트분밖에 없고, 특수용도의 호환제품에서 이 어드레스를 사용하고 있을 가능성은 매우 높으므로 만일 직접 제작하고자 할 경우 자기가 절대로 사용하지 않는다고 생각되는 별도의 어댑터보드가 사용하고 있는 어드레스에 할당하든가, 딥 스위치 등에서 자유도를 넓혀 설계하면 될 것이다.

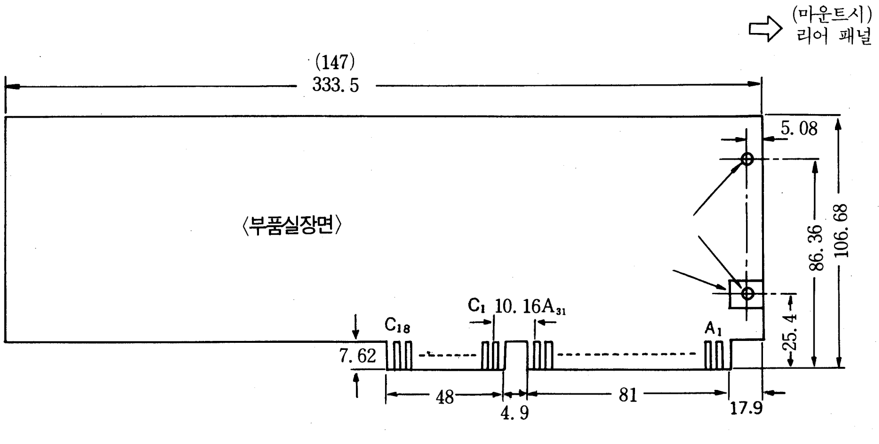
이것으로도 충분하지 않을 경우는 I/O공간을 증설하는 수 밖에 없다. 어댑터보드에서는 $A_0$~$A_9$를 디코드하여 칩 셀렉트신호를 만들어 내고 있고, 시스템보드에서는 $A_{10}$~$A_{15}$를 전부 무시하고 있으므로 I/O공간에는 1024바이트마다의 팬톰(이미지)이 63개 나타난다. 또 시스템 보드내의 각 I/O는 $A_8$과 $A_9$가 양쪽 모두 "L"레벨일 때 선택되도록 되어 있다. 즉 각 팬톰($A_8$, $A_9 \neq 0$)의 하위256바이트는 어느 보드도 사용하고 있지 않는 것으로 되어 있으며 AT에는 이러한 부분을 전부 합쳐 12,032바이트의 I/O공간이 한번도 쓰지 않고 남아 있는 것이다 (그림 16 참고).

제작보드 등에서 이것을 실현하기 위해서는 적어도 $A_8$, $A_9$가 "L"레벨에서 선택되지 않도록 하는 $A_8$~$A_{15}$의 풀디코드가 필요하다. 참고로 AT의 어댑터보드의 외형칫수를 그림17에 나타내어 둔다.

〔그림 16〕 PC AT의 확장 I/O 공간




〔그림 17〕 PC AT 어댑터 보드의 외형치수




---

〈용어해설〉

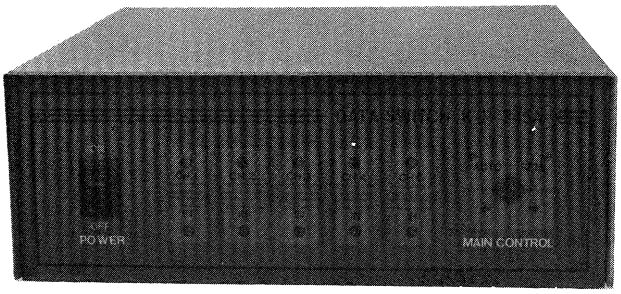
●게이트(Gate)
하나의 출력채널과 하나 이상의 입력채널을 가지고 있어 출력채널의 상태가 스위치회로의 과도상태를 제외한 그때의 입력채널상태에 의해 완전히 결정되는 논리소자를 말한다. 적어도 하나 이상의 입력채널을 가지는 논리소자의 조합이다.

●어드레스(Address)
번지라고 하며 어떤 단위의 정보를 기억하는 각 장소에 붙여진 식별번호를 말한다. 일반적으로 메모리속에서 정보의 하나의 단위(바이트라든가 워드)가 차지하는 특정한 장소를 지정하기 위하여 쓰인다. 이때 문에 번호를 가리키는 숫자로 나타내어지는 것이 보통이다.
이 숫자는 어떤 조작을 가하지 않고 그대로 어드레스로서 쓰이는 일이 있는데 이것을 절대 어드레스라 하며, 상대 어드레스는 어떤 수를 기점으로 하여 그 곳으로부터의 수로 나타내는 것을 말한다. 또한, 수대신 기호로 나타내는 것을 기호 어드레스라고 한다.

# PRINTER 주변기기 전문 메이커

- ●컴퓨터는 2대 혹은 3대이고, 프린터는 1대뿐이라 불편하게 사용하고 계십니까?
- ●컴퓨터 1대에 프린터를 1대씩 사용하거나 추가 구입시 프린터를 구매하려고 생각하십니까?
- ●프린터를 하고 있는 동안 컴퓨터를 장시간 사용하지 못하고 계십니까?
- ●CAD 사용시 Plotter가 그리고 있는 동안 컴퓨터를 장시간 사용하지 못하고 계십니까?
- ●컴퓨터의 수량과 프린터의 수량이 안맞아 적절한 사용을 못하고 계십니까?
- ●학교 및 학원의 실습실에 프린터대수가 부족해 적절한 교육을 못하고 계십니까?
- ●F.A 혹은 계측기쪽의 프린터 인터페이스에 문제가 있습니까?

## 컴퓨터 사용중에 불편한 위의 모든 문제를 자신있게 처리해 드리므로, 귀사의 업무에 도움이 되어 드리고 싶습니다!

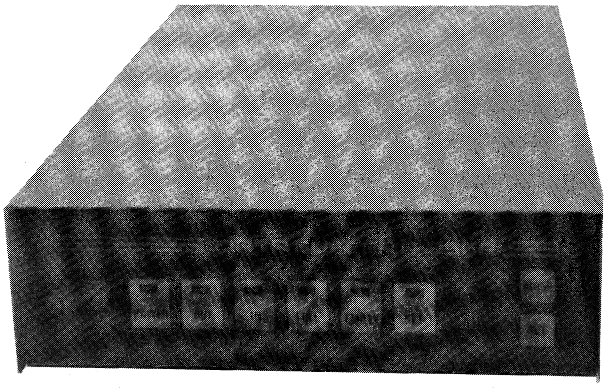



**K-P-345A의 사양**

| | |
|---|---|
| 입·출력 방식 | Centronics Type |
| 입력 COMPUTER | 3 : 1~5 : 1 |
| 출력 PRINTER | 1대 |
| 대기 시간 | 5초, 10초, 20초, 40초 중 선택 |
| 채널 변환방식 | 전자식 |
| 채널 선택방식 | 자동 or 반자동 |
| 용지 F.F 기능 | 자동 F.F or None F.F |
| F.F 삭제 기능 | 자동 삭제 or None |

**K-256P의 사양**

| | |
|---|---|
| 버퍼 용량 | 256kB(80col 약 200장) |
| 입·출력 방식 | Centronics Type |
| 입·출력 수 | 1 : 1 or 2 : 1 |
| 사용자 MODE | 10EA |
| 고유 기능 | Data 중 Space 입력 기능, 재 Copy 기능, 메모리 기능, Self Test 기능 등 총 10가지 |



## ● DATA SWITCH

- 전자식 완전자동 (2 : 1~8 : 1 or 64 : 1)
- 전자식 반자동 (2 : 1~8 : 1, 4 : 2)
- Serial 완전자동(4 : 1, 8 : 1)

## ● DATA BUFFER

- 소용량 64kB
- 고성능 다기능 256kB
- 대용량 다기능 1MB
- Serial용 1MB

## ● DATA CONVERTOR

- Serial→PARALLEL
- PARALLEL→Serial

## ● 각급 학교 및 학원의 전산 실습실 프린터 공유장치 공사


**ATTA SYSTEM**
PRINTER 주변기기 전문메이커
서울·종로구 종로 3가 175-4(현대APT709호)
전 화 : 2 7 7 - 1 7 3 9


● 상기 제품은 아래의 전문 취급점에서 전시 판매하고 있으며 직접 사용해 보실 수 있습니다.●

| | | |
|---|---|---|
| 서울 : | 세운상가 협진 | 274—5419(가동 라열 426호) |
| | 용산 나진상가 동해전자 | 719—8010(17동 가열 304호) |
| | 용산 전자랜드 협진 | 712—2764(3층 C-336호) |
| 인천 : | 바이너리 컴퓨터 | (032)762—5544 |
| 대구 : | 범진 컴퓨터 | (053)252—2211 |
| 마산 : | 한길 컴퓨터 | (0551)94—5555 |

(기타 지역 전문 취급점 모집중)



# IBM PC의 국내 현황

## 재도약의 시점에 선 XT

컴퓨터의 역사가 그리 길지않은 국내에 81년부터 컴퓨터 붐을 일으켰던 8비트 PC의 아성속으로 NEC5200과 IBM 5550이 소개되었다. 이것이 국내에 16비트 PC가 들어오게된 계기가 되었다. 그러나 이때는 고가의 제품이라는 점과 호환성의 문제점 때문에 컴퓨터 사용자에게는 약간의 거리감을 주었다.

이러던 차에 84년에 삼보 컴퓨터, 삼성전자 등 몇몇 업체들이 IBM PC XT와 호환성을 가지는 제품을 개발해 내면서 가격의 하락과 함께 점차 보급이 늘어나고 8비트 PC에서 16비트 PC로의 전환이라는 새로운 컴퓨터 시장의 전환점을 마련하였다.

그러나 XT가 사용자층 형성을 위한 자리를 굳히기 전에 AT와 386의 등장으로 그 방향을 잃어가는 경향을 띠지만 16비트 PC를 가지고 있는 사용자들이 대부분 워드프로세서나 OA용으로 사용하고 있고 이러한 업무는 XT로도 충분히 처리할 수 있다는 점과, 89년부터 시행되는 전산화 교육정책에 따라 중·고·대학생들 사이에서, PC를 처음 다루는 사용자들에게서, 그리고 정보화사회의 터미널 역할을 담당하기에 적합한 필수품으로서 계속적인 인기를 유지할 것으로 보이며 이에 따라서 국내 20여개의 XT 개발업체는 가격을 인하하면서 재도약을 위한 새로운 시도를 모색하고 있다.

## 국내 제품 현황

**●명필 XT**

적용 업무의 전산화, 사무 자동화, 중·대형 컴퓨터의 온라인 단말기로 광범위하게 이용되는 다목적 보급형 시스템인 고려씨스템의 명필 XT는 8MHz의 8088-II를 마이크로프로세서로 채택하였고 주기억 용량이 512kB로 최대 640kB까지 확장가능하며 OS로 MS-DOS를 사용하고 있다.

동제품의 특징은 고해상도의 칼라그래픽 기능이 있어 각종 설계업무 및 특수분야의 그래픽 처리에 용이하다. 또한 IBM, UNISYS, FACOM, VAX, PRIME 등의 기종과 연결하여 명필 XT 상호간의 네트워크나 LAN과 LAN의 인터코넥션, 커뮤니케이션 서버 등을 통해 토탈 네트워크화를 꾀할 수 있는 터미널 에뮬레이션 기능이 있어 별도의 터미널 기능이 필요없다는 것이다.

동제품은 개인업무자(교수, 변호사), 일반기업과 연구기관, 관공서, 학교를 대상으로 판매되고 있으며 사용자에 대한 사전 사후 교육지원 및 정비보수를 위해 전국 대리점망을 통해 통합지원 체제를 마련하고 있다.

**●트라이젬 POP**

88년 10월에 (주)삼보 컴퓨터에서 출하된 트라이젬 POP은 데스크탑형의 16비트 OA용 컴퓨터로 국내의 XT 기종 중에 가장 빠른 10MHz의 처리속도를 가지고 있다.

동제품은 원터치의 전면 파워스위치, 리세트스위치, 스피드 인디케이터, 키록(keylock), 리얼타임 클럭 기능을 내장하여 축소 집약의 소형경량화를 실현하였고 인텔 8088의 마이크로프로세서를 사용하였다.

주기억용량이 640kB, 외부 기억장치로 360kB의 FDD 5¼인치 1개 또는 2개, 20MB의 HDD, 84키의 한글/영문 키보드, 5개의 확장 슬롯으로 구성된 동제품은 129만원과 174만원의 가격으로 일반 OA업무 및 워크스테이션용으




로 판로를 형성하고 있다.

●마이티 16-Ⅲ

(주)금성사는 마이티 16-Ⅱ에 이어 인텔리전트터미널 기능을 중·대형 컴퓨터와 한글 제닉스에서 구현할 수 있는 마이티 16-Ⅲ를 종전의 마이티 16-Ⅱ보다 저가격으로 88년 9월부터 XT 시장에 선보였다.

마이티 16-Ⅲ는 IBM, UNIVAC, VAX, PRIME, MV시리즈, HP, CDC와 금성의 중·대형 컴퓨터인 히다찌 M시리즈, DPS-678, GS-8000, ALTOS, 3B 등과 접속하여 한글 터미널 기능을 효과적으로 수행하도록 되어 있다.

특히 토큰-링 방식의 GS-TRN과 접속하여 기존의 LAN 개념을 수용하고 초고속의 전송속도와 대용량(80MB/140MB)의 하드디스크 공유를 구현하여 PC의 사용자에게 문제가 되어온 대용량(140MB 이상)의 화일을 구축할 수 있도록 하고 한개의 화일을 LAN상에서 72명의 사용자가 동시에 사용할 수 있는 특징을 가지고 있다.

현재 129만원과 174만원(20MB의 하드디스크를 부착했을 경우)으로 고정되어 판매되고 있는 마이티 16-Ⅲ는 전문 사업장인 회계/세무 사무실, 의료보험조합, 치과 등의 단과병원, 운수회사가 주된 판매대상이 되고 있다.

●포커스-100

일진전자는 기업체 관공서, 공장, 병원, 학교, 연구기관 등의 종합 사무자동화 시스템에 적합하도록 설계한 포커스 100을 87년부터 출하하여 판매하고 있다.

한글 한자코드의 국가 표준안을 국내 최초로 적용시켜 한글 2319자, 한자 5500자, 심볼 564자, 사용자 정의글자 500자를 사용할 수 있는 포커스-100은 동사의 옵니네트 LAN의 워크스테이션으로 사용할 수 있어 각종 디바이스, 프로그램 데이터 화일을 공유할 수 있고 호스트와 PC, 호스트와 LAN, 패킷 교환망(X.25), 브로드 밴드 등의 LAN과 LAN의 인터페이스에 이용되고 있다.

또한 리세트 키, 파워스위치, 키보드 코넥터를 전면에 부착하고 배터리 백업 장치를 통한 리얼타임클럭 기능을 내장하였고 인사관리, 회계관리, 판매관리, MRP, 병원업무, 건설공정관리, 중소기업 토탈시스템, 보급소 업무에 관한 소프트웨어를 지원받을 수 있다.

현재 포커스 100은 전국 주요도시의 A/S지사망을 통해 정기순회 점검과 동사의 기본교육 및 전문교육을 통해 원활한 사용을 돕고 있다.

●CPC-4000

(주)대우전자에서 88년 1월에 고성능 저가격으로 XT시장의 활기를 찾고자 출하한 CPC-4000은 인체공학 시스템에 의해 설계된 다양한 기능의 키와 사용자의 위치에 따라 키보드 및 모니터를 자유롭게 이동할 수 있어 좁은 공간에서도 설치가 쉽도록 한 것이 특징이다.

동제품의 특징은 한글 MS-DOS용(한글 MS-DOS, DH한글) 및 표준한글용(표준한글, DH한글)을 병렬 선택하여 사용자 선택에 따라 편리하게 사용할 수 있다.

또한 배터리 백업된 클럭/캘린더 기능을 가지고 있어 정전시 타임키핑(Time Keeping)이 가능하게 하였고, MDA, 허큘리스, CGA, AT&T모드 기능이 원보드상에 내장되어 있어 모니터 선택에 따라 다양한 비디오 모드를 지원한다. 속도는 키보드에 의해 4.77/8/10MHz를 필요에 따라 변환할 수 있고 시스템의 속도를 표시하는 LED가 전면에 부착되어 현재의 속도를 알 수 있다.

현재 CPC-4000은 180만원에 CPC-4020은 260만원으로 사무자동화 제품으로서 일반 중소기업에 판매되고 있다.

●프로-2000

대우통신이 개발하여 모델-D라는 상품명으로 미국시장에 수출하면서 50만대를 판매하게된 프로-2000은 가격을 인하하면서 국내 및 세계시장 확대에 박차를 가하고 있다.

프로-2000은 VLSI를 사용하여 회로를 커스텀화 하여 PCB의 사이즈를 ½(타기종의 ⅙)로 컴팩트화 하였고 보조기억장치의 슬림타입을 채택하였으며 각종 인터페이스를 위한 RS-232C, 병렬포트, 허큘리스 모노 그래픽카드, 칼라어댑터를 내장하여 마더보드의 공간에 여유를 두어 별도의 메모리보드의 추가없이 적당한 IC의 추가로 640kB까지 기억용량을 늘릴 수 있고 슬롯은 4개 내장되어 있다.

동제품은 섬세한 해상도(720×348픽셀, 640×200픽셀)와 16색 칼라그래픽 처리기능을 보유하고 있고, 별도의 보드없이 모노, 칼라 모니터를 바꾸어 사용할 수 있어 고해상도의 텍스트와 그래픽이 유지되는 특성을 가지고 2개의 FDD를 포함하여 135만원, 20MB의 하드디스크와 1개의 FDD를 포함하여 192만원으로 현재 판매되고 있다.

**●슈퍼-16E**

현대전자는 전산화 교육정책에 따라 교육용 컴퓨터 시장을 넓히고자 슈퍼-16E 판매를 강화하고 있다.

슈퍼-16E는 컴퓨터 교육과 학습, 생활, 업무 등에 이용할 수 있어 초·중·고생 및 대학생, 일반인까지 온가족이 사용할 수 있는 장점을 가지고 있다.

동제품의 특징은 정부표준 워크스테이션과 동일한 한글 키보드(86키)를 채택하고, 디스크 장치를 내장하여 사용이 용이하도록 하고 FDD가 소형화되고 있는 추세에 따라 3.5인치 FDD의 장착이 가능하도록 한 것이다.

슈퍼-16E는 비디오 모드에 한글 완성형 카드, MVA카드, EGA카드가 있느냐에 따라 모델명이 A, B, C로 구분된다.

응용분야는 CAI시스템에 의한 교육용 소프트웨어를 지원하여 효과적인 컴퓨터 교육을 실시하고 있으며 한글 DOS 및 MS-DOS를 사용할 수 있고 LAN을 구축하여 천리안 및 한경 KETEL DB 정보를 검색할 수 있는 터미널로도 사용할 수 있다.

현재 슈퍼-16E의 A모델은 85만원에 B모델은 74만원에 시판되고 있다.

**●SPC-3000B**

삼성전자는 업무용 컴퓨터로 SPC-3000A에 이어 SPC-3000B를 개발하여 시장에 내놓았다.

동제품은 클럭속도를 높이기 위해 10MHz의 인텔 8088을 CPU로 채택하면서 주기억 용량을 512kB에서 최대 768kB까지 확장가능케 했고 각종 중·대형 컴퓨터의 워크스테이션용으로 사용하기 위해 소프트웨어가 지원되어 OA용 독립시스템 및 한글 온라인 단말기용, 특수 전문분야 등에 응용할 수 있다.

또한 OA-II 글벗, OA-II 셈벗, DBMS의 OA전용 패키지와 베이직, 코볼, 포트란, C, 파스칼, 어셈블러 등의 프로그래밍 언어 및 400여종의 업종별 응용 소프트웨어를 사용자의 업종과 회사의 특성에 따라 지원받을 수 있다.

현재 SPC-3000B의 SG모델은 131만원, 2G모델은 186만원에 판매되고 있다.




## 최근 서서히 부상하는 AT

요즈음 AT란 말을 주위에서 흔하게 듣게 되었다. IBM PC/AT와 호환기종들을 통틀어 총칭하는 의미로 널리 쓰이게 된 것으로 국내에서는 1985년 삼보컴퓨터에서 인텔의 80286프로세서를 탑재한 트라이젬 AT가 발표되면서 처음으로 소개되었고 그 다음해인 86년부터 AT호환기들이 대거 선보이기 시작했다.

비록 초창기에는 가격이 비교적 고가였기 때문에 누구나가 쉽게 접할 수는 없었지만 불과 몇년이 지난 오늘에 와서는 고기능 저가격화의 실현으로 기존 XT의 시장을 크게 잠식하였으며 심지어 XT위기설까지 나돌게 하였다.

이와 같은 현상은 과거 IBM PC XT와 그 호환기종들이 8비트 퍼스컴들을 교육용 또는 제어용 등 별개의 시장으로 밀어내고 PC시장을 금새 장악했던 양상과 흡사하다.

**●가격문제**

국내 AT기종의 꾸준한 시장성 창출을 가능케한 요인중 한가지는 AT시장에 참여하는 업체들의 급증으로 인한 수요창출과 컴퓨터 본체규모를 콤팩트화시킴으로써 얻는 비용의 절감을 들 수 있다. 현재 시중에서 판매되고 있는 AT기종의 가격은 150만원에서 450만원대에 이르고 있는데 이 가격폭은 2년전의 500만원에서 700만원대란 가격폭과 비교해 볼 때 상당한 변화를 보여준다. 다음은 AT가 국내에서 대두되기 시작했던 86년부터 올해까지의 가격변동을 주요 모델들과 함께 나타낸 것이다(표 1,2,3 참고). 그리고 여기에 명시된 가격은 각 업체에서 권장하고 있는 시스템구성 내역에 따라 20만원 정도의 변동폭을 지니며 자료는 각 업체에서 제공받은 것임을 밝혀둔다(가격 미정인 경우는 제외했음).

**●성능문제**

AT가 PC시장을 넓혀갈 수 있었던 또하나의 요인으로 XT에 비해 AT가 갖는 성능의 우수성(표 4 참조)을 들 수 있다. AT는 OS로서 MS-DOS는 물론 제닉스를 탑재할 수 있고 제닉스 탑재시 멀티유저용으로도 사용가능하며 주프로세서로는 인텔의 80286과 옵션인 연산보조용 프로세서 80287을 갖출 수가 있다. 또한 고해상도 그래픽기능

을 제공, CAD/CAM 등의 워크스테이션용으로도 가능하다는 점이 유저들의 구매욕을 불러일으키기에는 충분했던 것으로 본다.

이러한 성능의 대폭적인 향상은 근본적으로 80286 마이크로프로세서로부터 나온다. 현재 16비트 CPU시장에서 압도적인 우위를 점하고 있는 이 프로세서는 새로운 VLSI로서 고급의 어플리케이션, 즉 비지니스컴퓨터, 워크스테이션, 통신시스템과 같은 멀티유저 멀티태스크시스템에서의 응용을 목표로 고안된 것으로 이점이 저가격과 집적도 향상을 꾀했던 80186프로세서와 다른 점이다.

여기서 잠시 인텔의 마이크로프로세서 계보(특집 I 참조)를 살펴보면 8086과 80286, 그리고 80386과의 계보연계성 및 그 호환정도를 쉽게 파악할 수 있다. 즉 인텔은 유저들의 기대에 부응하여 8086의 후계로 80286을 개발하였고 그 기능들을 또한 32비트 CPU인 80386과도 호환을 유지하게 하여 80286에 대한 유저의 구매결정이 결코 헛되이 되지 않게 고려하고 있다는 점 또한 AT의 성장에 적지않은 영향을 미쳤을 것으로 본다.

설계구조도 8086과의 호환을 고려하여 8086과의 호환모드인 실주소모드방식과 80286 자체의 고급기능인 보호모드 등 2개의 동작을 구비하고 있어 시스템 설계자가 필요에 따라 하나를 선택할 수 있게 하였다.

실주소모드방식은 8086과의 호환을 염두에 둔 것으로 일부 명령어의 삭제, 수정을 통해 8086의 소프트웨어를 쉽게 실행시킬 수 있다(이것은 8086 CPU에서 축적된 방대한 양의 소프트웨어를 계승하기 위한 조처이다).

80286의 또다른 모드인 보호모드는 실주소모드에서 CPU의 상태제어 레지스터인 MSW의 PE비트를 세트함으로써 설정되는데 바로 이 경우, 메모리관리, 보호관리 및 태스크관리 등이 활성화된다. 그리고 이 방식(보호모드)에서도 8086CPU와의 호환성은 사라지는 것이 아니며 단지 소스프로그램 수준에서 다시 어드레싱을 통해서만이 가능해지는 불편만이 따를 뿐이다(8086 CPU에 어드레싱이 편성돼 버린 실행소프트웨어는 불가능함).

마이크로프로세서 기능의 발달과 함께 IBM측은 기본 관리방침인 대형컴퓨터의 고급소프트웨어를 축소, 재구성하여 PC수준에서도 훌륭하게 실행시킬 수 있도록 지원하면서 비교적 고급의 기능들을 유저에게 지원하게 되었다. 멀티태스크기능과 멀티유저기능이 그 대표적인 예인데 이는 앞에서 설명했던 자원의 공유, 보호관리, 태스크관리 등의 새로운 문제들을 해결해야만 가능한 것이다.

자원 공유의 문제는 여러가지 작업

〔표 1〕 86년 주요 AT기종 가격 비교

| 업 체 | 모 델 | 가 격 |
|---|---|---|
| 금성반도체 | TCP 1280 W/S | 1천달러 |
| 고려시스템 | 고려 AT 200 | 6백70만원 |
| 동양정밀 | OP-COM/AT | 7백만원 |
| 삼보컴퓨터 | TriGem-AT | 5백30만원 |
| 삼성전자 | MFC-5000 | 5백만원 |
| 삼성반도체 | OX-85 | 1천~2천달러 |
| 삼성HP | VECTRA | 6백50만원 |
| 선진컴퓨터 | SYMBIOS AT | 4백90만원 |
| 썬트로닉스 | SUPER AT | 7백만원 |
| 일진전자 | FOCUS-16AT | 5백만원 |
| 제우스컴퓨터 | EXP | 5백90만원 |
| 태일개발 | TIDCOM-AT | 5백50만원 |
| 텔레비디오 | TELE-AT | 5백만원 |
| 해동시스템 | TOMCAT-AT | 5백60만원 |
| 홍익전자 | SUPER S/EC-3052 | 5백50만원 |
| 한국IBM | PC-AT | 3천9백95달러 |

〔표 2〕 87년 주요 AT 가격 비교

| 업 체 | 모 델 | 가 격 |
|---|---|---|
| 금성사 | 마이티 AT | 5백50만원 |
| 고려시스템 | AT 200 | 5백60만원 |
| 동양정밀 | OPC-286 | 3백만원 |
| 삼보컴퓨터 | 트라이젬 AT | 4백99만원 |
| 삼성전자 | MFC-5000 | 6백76만원 |
| 삼성HP | VECTRA | 6백50만원 |
| 선진컴퓨터 | SYMBIOS AT | 4백98만원 |
| 썬트로닉스 | OA-16 수퍼 AT | 4백만원 |
| 제우스컴퓨터 | EXP | 5백40만원 |
| 태일개발 | TIDCOM AT | 6백30만원 |
| 텔레비디오 | TELE-AT | 5백만원 |
| 해동시스템 | TOMCAT-AT | 5백만원 |
| 홍익전자 | SUPER 5 | 5백5십만원 |
| 한국IBM | PC-AT | 6천6백61달러 |
| 동서전자공업 | 엑심 16V-20 | 1백40만원 |
| 삼창전자 | 사이보그 AT | 4백만원 |
| 콤텔 | CTL/AT | 3백50만원 |
| 유니온컴퓨터 | 유니온 AT | 3백50만원 |

〔표 3〕 88년 주요 AT기종 가격 비교

| 업 체 | 모 델 | 가 격 |
|---|---|---|
| 뉴텍코리아 | WALCOM | 3백63만원 |
| 고려시스템 | 고려 286 | 3백65만원 |
| 삼보컴퓨터 | 트라이젬 286+ (BDA5) | 3백80만원 |
| | 트라이젬 286Ⅲ (BDA5) | 4백만원 |
| 제우스컴퓨터 | EXP/10 | 3백만원 |
| 홍익컴퓨터 | SUPER-AT (Turbo) | 2백15만원 |
| 한국IBM | 모델 30-286 | 2백~3백만원 |
| 일진전자 | FOLUS-200 | 3백10만원 |
| 메가컴퓨터 | 메가-286AT | 3백30만원 |
| 큐닉스 | 7700 | 4백20만원 |
| 현대전자 | SUPER-16E | 1백89만원 |
| SAM KOREA | SAM-AT | 2백만원 |
| 제일나노텍 | NANO 286A-10 | 2백10만원 |
| 한국마이크로시스템스 | P-3204 | 3백50만원 |

〔표 4〕 XT와 AT 호환기종의 주요 규격 비교

| 내부 통신방식 | 16비트(XT) | 16비트(AT) |
|---|---|---|
| 마이크로프로세서 | 8088, 8086 | 80286 |
| 연산 보조용 프로세서 | 8087 | 80287 |
| 클럭 속도 | 4.77/8MHz | 8/10MHz |
| 주기억용량 | RAM : 512/640KB | 640KB, 1MB |
| 보조기억장치 | FDD : 0.5MB<br>HDD : 20/30MB | 1.2MB<br>20/30/40/80/140MB |
| 확장 슬롯 | 8 | 8 |
| OS | MS-DOS | MS-DOS, XENIX |

(task)을 동시에 처리하기 위해서 메모리 자체가 I/O장치 등 시스템의 자원을 공유해야 하는 것으로 80286프로세서는 매핑테이블(Mapping Table)을 이용하여 세그먼트의 관리는 물론 물리어드레스 공간과 가상어드레스 공간을 분리하여 메모리 공유를 실현하고 있다. 80286프로세서의 이러한 메모리관리의 결과 프로세서가 액세스할 수 있는 물리어드레스 공간도 8086에서는 1MB였던 것이 80286에서는 16MB로 확장되었고 가상적으로는 한개의 태스크마다 1GB의 주소공간이 존재하고 있는 것처럼 다룰 수 있게 되었다.

두번째의 문제인 보호관리는 시스템내에 복수개의 프로그램이 동시에 존재하기 때문에 나타나는 것으로 이들 프로그램 상호간의 배타적 관계(독립성 유지)를 실현하기 위한 보호처리 및 관리의 문제이다. 80286프로세서의 보호처리 메카니즘을 보면 하드웨어가 최대 4개의 Privilege Level을 각 태스크에 제공하여 시스템의 계층화를 실현하고 있으며 하드웨어에 의한 메모리관리 기능을 이용하여 태스크간의 어드레스 공간을 격리시켜 태스크 또는 유저간의 영향을 배제할 수 있게 되었다. 그리고 이들 보호기능들을 모두 마이크로프로세서에 집적시킴으로써 고속처리를 실현하였다.

다음은 여러개의 태스크를 효과있게 실행하기 위한 실행우선 순위를 결정하고 관리해야 하는 태스크관리의 기능으로 80286프로세서에서는 태스크관리를 지원하는 기능도 하드웨어로 실현하고 있어서 모든 태스크들은 다른 태스크의 간섭으로부터 빠른 속도로 상호배제를 유지하게 되었다.

또한 속도에 있어서도 AT는 CPU내에 4개의 독립된 처리장치의 구현으로 데이터 버스 요구의 극소화와 CPU 처리속도의 극대화를 꾀했다. 이들 4개의 유니트는 파이프라인(Pipelined)화 되어 동작하기 때문에 놀라울 정도의 처리속도를 지니게 되었다.

## ●국내 제품 현황

현재 국내에서 AT호환기종의 시장에 참여하고 있는 업체들은 30여개사에 이르고 있어 20여개사 수준이었던 작년에 비해 현저한 증가를 보여주고 있다.

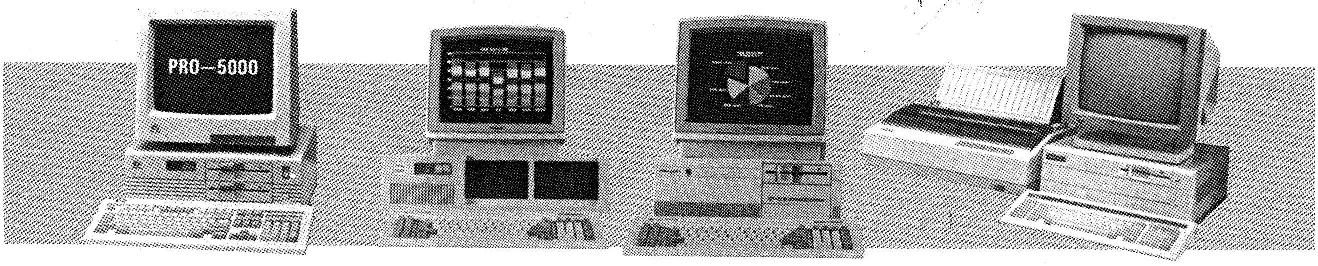
AT가 국내에 대두되기 시작했던 86년부터 한국IBM, 삼성반도체통신, 삼성전자, 테레비디오코리아, 제우스, 삼보, 고려, 해동, 홍익, 동양정밀, 썬트로닉스, 금성사 등 10여개사가 PC/AT 시장을 꾸려가기 시작했으며 그후 계속되는 업체들의 참가로 AT호환기들은 저가격화와 함께 인기를 계속 누려왔다.

### ●TRIGEM286-Ⅲ

올해 PC수출실적 1억불을 달성한 삼보컴퓨터인 경우 트라이젬 286+와 286Ⅲ를 선보이고 있는데 286Ⅲ는 AT 호환기종중 상위컴퓨터에 속하며 OA업무 및 일반 엔지니어링용으로 적합하다. 두 기종 모두 640kB의 RAM과 1.2MB 용량의 5.25인치 FDD, 20/40/80MB의 HDD를 장착하고 있으며 OS로는 MS-DOS, XENIX를 지원하고 있다. 그리고 트라이젬 PC를 이용해 동일 건물내의 사무실마다 분산운영되어 서로의 PC 및 주변기기를 공유케 하기 위한 강력한 네트워크인 LAN시스템은 •편리한 풀다운 메뉴드라이버방식 •완벽한 프로텍션 시큐리트기능 •2바이트 KS완성형, 2바이트 조합형 등 모든 한글, 한자사용 •Hot Key기능에 의한 신속한 모드 변환 •On Line Help기능 등을 구축해 주게 된다.

### ●EXP/10

PC국산화의 산기지로 알려진 제우스컴퓨터인 경우 EXP에 이어 EXP/10




을 자체개발 생산하고 있는데 이 기종은 10MHz의 CPU와 1.2MB 용량의 5.25인치 FDD, 280MB(140×2)의 5.25인치 HDD를 탑재하고 있으며 한글 CRT는 9대(최대 16대)까지 연결가능하며 기타 장치로서 터미널, 모뎀, 버퍼, RS-422 등이 있다.

**●WALKOM**

뉴텍코리아의 WALKOM은 랩탑형 컴퓨터로 시대감각에 맞는 디자인으로 시간, 장소에 구애받지 않고 사용할 수 있다는 점이 특이하다. 이 WALKOM은 1MB의 RAM, 1.44MB(3.5인치) FDD, 20MB의 HDD를 지니며 이외에도 HDD OFF시 4시간동안 사용할 수 있는 충전식 배터리와 표준 I/O포트로서 EGA/MGA/CGA 모니터 포트를 갖추고 있다.

**●PS/2 30-286**

9월에 한국IBM에서는 IBM PS/2의 최소기종인 30모델의 더욱 강화된 30-286모델을 발표함으로써 IBM PS/2계열의 선택 폭을 넓히게 되었다. PS/2 30-286모델은 10MHz의 80286프로세서를 탑재하여 기존의 30모델에 비해 약 2배의 빠른 속도를 제공하고, 그래픽기능에는 과거 상위모델에서만 제공되던 VGA(Video Graphic Array)기능을 제공한다.

그리고 기존 PS/2 30모델 용량의 약 25배에 해당하는 최대 16MB까지의 메모리 확장성을 제공하며, 기존 30모델의 2배 용량으로 3.5인치 고밀도(1.44MB) FDD를 내장하고 있다. 또한 최근에 IBM에서 발표한 MS-DOS4.0 등의 OS와의 조화는 각종 업무에 있어 훌륭한 성능과 잇점들을 사용자에게 제공하리라 보고 있다.

**●큐닉스 모델 7700**

특히 개인용 컴퓨터로서 대형컴퓨터의 단말기 기능 수행, 스테이션간의 네트워크기능이 합쳐진 시스템들이 많이 발표되었는데 큐닉스의 7700모델을 보면 SNA/SDLC 통신카드, COAX인터페이스 카드, TWINAX인터페이스 카드, LAN인터페이스 카드 등을 지원하여 IBM대형기종을 포함한 컴퓨터시스템의 온라인 단말기로 사용할 수 있으며 LAN기능과 함께 워크스테이션의 필수적인 소프트웨어인 워드프로세서, 스프레드 시트, dB 등을 지원하고 있다.

**●고려 286**

마찬가지로 IBM대형기종의 단말기능 및 통신기능으로 사용가능한 고려시스템의 고려 286은 MS-DOS, XENIX 및 UNIX OS를 채택하여 싱글과 멀티유저를 선택하여 사용할 수 있으며 IBM PC에 사용중인 모든 패키지를 실행시킬 수 있고 IBM PC/AT, XT에서의 완벽한 하드웨어 호환으로 모든 옵션 PCB의 접속을 허락한다.

특히 이 기종에서는 4개 LED지시자로 자기진단기능을 갖고 있어 앤드유저가 직접 시스템을 체크해 볼 수 있으며 PROM 메모리가 64kB로 32kB는 BIOS로 사용되며 32kB는 유저가 사용할 수 있게 하였다.

이외에도 최고 15MHz의 속도와 1MB의 RAM을 지니는 홍익전자의 수퍼-AT(Turbo), 한/중 카드 및 허큘리스, EGA, 각종 S-EGA 카드 등을 옵션으로 제공하여 디스플레이에 역점을 둔 제일나노텍의 NANO-286A-10, 컴팩트한 사이즈에 모노칼라 콘트롤러를 내장하고 768kB의 메모리를 갖추고 있는 한국마이크로시스템스의 P-3204모델, 동인무역의 HOT-LINE 286, 하나컴퓨터에서 OA워크스테이션용으로 발표한 TOP-286 등 AT시장을 향한 신모델들이 점차 늘어나고 있으며 판매활동도 활발히 진행중에 있다.

## 가격문제로 침체의 늪에 빠진 386

지난 86년말 미국에서 개최된 컴텍스쇼에서 컴팩(Compag)사가 데스크프로 386을 처음 선보인 이래, 국내업체에서도 32비트 PC시장의 참여에 관심을 갖기 시작하였다.

87년 전자쇼 이후 본격 출하하기 시작한 386PC는 기존의 AT기종에 비해 향상된 스피드와 멀티유저, 멀티태스킹을 무기로 하여 PC기종 가운데서 최상위 제품으로 등장하였다.

현재, 그룹계열사를 끼고 있는 PC업체들을 포함, 상당수의 기업체가 386시장에 참가하고 있다. 국내시장에 나와있는 386시스템으로는 금성사 마이티 386, 삼보컴퓨터 TG386, 삼성전자 SPC6000, 삼성 HP 벡트라386, 유스텍 WYSE PC 386, 에이스컴퓨터 수퍼 386, 조우니컴퓨터 Jyoni 32F, 고려시스템 고려 386, 쌍용컴퓨터 Dragon386, 대우통신 PRO-5000, 엘렉스컴퓨터 매킨토시II, 동양정밀 OPC 386, 테레비디오 TeleSTAR 386, 케스타 수퍼 386, 삼일컴퓨터 ExTRA 386 등 다수이다.




대개 이러한 386기종은 I80386계열 중의 하나인 컴팩이 주종을 이루고 있다. 국내 주요업체들이 컴팩을 채택하고 있는 이유로는 IBM XT/AT와의 호환성과 기존의 XT, AT에 사용하던 DB나 로터스 1-2-3 등 S/W를 활용할 수 있다는 점이다.

현재 32비트 PC에 사용할 수 있는 OS로는 UNIX, XENIX, PC-MOS/386, THEOS, Window/386, MERGE 386, VM/386, DESQView2.00, FLEX OS 386 등 15개 정도가 넘지만, 국내의 386PC는 통합 OS가 개발이 안된 상태이기 때문에 뚜렷한 통합 OS가 없이 UNIX, ZENIX, MS-DOS 등을 개별적으로 수용, 지원하고 있다.

## 국내 제품 현황

**●마이티 386**

모델명 GMC8040이라고 불리우는 금성사의 마이티 386은 인텔 80386을 채용하여 처리속도를 AT보다 4배나 향상시킨 제품으로서 보조 기억장치로 1.2MB의 5.25인치 FDD와 80MB까지 확장가능한 하드디스크 및 고해상도(640×400)비디오 카드, 그래픽 카드 등을 내장하고 있어 각종의 CAD/CAM 작업을 효율적으로 처리할 수 있다.

또한 호스트의 커뮤니케이션 및 네트워크 서버기능을 가지고 있어 상위기종과 하위기종 사이의 네트워크 구성이 용이하며, 워드프로세서 패키지, 데이터베이스 통합 패키지 등을 사용하여 OA분야에도 사용할 수 있다. 한편 운영체제로는 MS-DOS3.2와 제닉스를 수용하고 있다.

**●TG386**

삼보컴퓨터의 TG386은 주기억장치에 롬64kB, 램2MB이며 비디오 램은 64kB로 기억용량이 크다. 특징으로는 CPU보드가 마더 보드형이 아닌 IBM 옵션 카드 형식인 카드 타입으로 슬롯을 꽂아 사용할 수 있도록 하였고, ROM-BIOS로는 어워드(Award)를 채택하고 디자인설계에 소요되는 시간적 손실을 줄이기 위해 Chips & Technology사의 커스텀IC와 회로도를 사용한 것으로 알려져 있다. 동사에서는 중소기업 OA용 외에도 고도의 엔지니어링 연구개발을 위한 CAD/CAM, FA분야 등 특수분야의 업무에 쓰이며, 멀티유저 및 LAN구축을 실현하였다.

한편 삼보의 TG386V는 최근 개발된 32비트 PC로 20MHz의 초고속 처리와 폭넓은 시스템 확장성, 고도의 멀티유저 기능 등을 이용하여 고차원의 종합 OA업무, 과학기술분야, 엔지니어링 등 특수 분야의 업무에 활용할 수 있다.

**●SPC600**

삼성전자의 SPC6000은 86년 10월에 개발해 낸것으로 주기억용량 1MB를 표준으로 10MB 정도까지 확장 가능하다. 또한 동시스템은 IBM PC XT/AT 호환기종으로 기존의 16비트 퍼스컴의 소프트웨어를 사용할 수 있다. 또한 Shadow 램을 채택하여 스피드 업이 되고, 20MHz 클럭을 사용하기 때문에 빠른 속도를 가지고 있다. OS로는 MS-DOS, UNIX를 수용하고, 과학 기술용 및 CAD/CAM, 엔지니어링분야에 적합한 기종으로 데스크탑 퍼블리싱용 및 단위부서 업무 전산화용으로도 사용이 가능하다.

**●고려 386**

고려시스템의 고려 386은 지난 87년 3월에 출하되었고 메모리는 기본 1MB에서 최대 16MB까지 확장해서 사용이 가능하다. 한편 BIOS 및 EGA BIOS 등의 롬프로그램을 램으로 옮겨 사용하게 하므로서 처리속도를 향상시켰으며 한글처리에서는 국가표준코드(완성형), IBM조합형 등을 제공한다. 한/영 컬럼수가 다양하여 호스트 에뮬레이션, W/P 및 일반 유저프로그램 등 업무특성에 맞는 화면 구성이 용이하며, MS-DOS, OS/2, XENIX시스템V 386/286, Vp/ix(XENIX에서 MS-DOS 사용) 등 다양한 OS를 제공한다.

**●매킨토시 II**

미니급의 수퍼 개인용 컴퓨터를 지향하는 엘렉스 컴퓨터의 매킨토시 II는 국내의 80386계열과는 달리 모토롤라의 MC68020을 마이크로프로세서로 채택하고 주기억용량은 1MB를 표준으로 최대 8MB까지 확장이 가능하다.

동사가 자랑하는 S/W인 DTP, DTM, DTC, DTE 등은 기존의 W/P, 로터스 1-2-3, DB, SPSS에서 느껴보지 못한 컴퓨터의 묘미를 느끼게 해 준다는 것이 특징으로 DTP는 인쇄에, DTM은 OA, DTC는 커뮤니케이션, DTE는 아트워크에서 사용할 수 있다.

또한 동사에서는 한글 Talk를 발표, 국내 사용자들에게 매킨토시용 영문 S/W에서 별도의 수정없이 완벽한 한글처리가 가능하게 하였다. 그리고 Mac OS뿐만 아니라 A/UX와 MS-DOS 등의 다양한 OS가 제공되며 1680만가지의 색을 고해상도로 표현할 수 있다.

**●PRO-5000**

국내 가장 최소형을 자랑하는 대우통신의 PRO-5000은 지난 87년 3월경에 출하되고 인텔 80386마이크로프로세서를 채택하여 AT기종에 비해 2~3배, XT기종보다 무려 11배의 업무처리 속




도가 빨라졌으며 고속 ESDI의 80/160 MB 하드디스크 이용이 가능하다. 또한 MS/OS2, XENIX 386, 유닉스 시스템 등의 OS탑재가 가능하며, IBM 2바이트, 정부행정전산망 2바이트 완성형 및 7비트 완성형이 하나의 카드로서 처리 가능한 완벽한 한글/한자 카드를 제공한다.

또한 PRO-5000은 네트워크에 강하고 CAD쪽으로도 강해, 전자, 전기, 기계, 토목, 정밀건축 등의 설계분야에도 그 진가를 발휘할 수 있다.

**●드래곤 386**

쌍용컴퓨터의 드래곤 386은 IBM3270 에뮬레이터를 사용하여 IBM3090, 308x, 43xx, 9370 등 대형컴퓨터의 온라인 단말기로 사용할 수 있으며, 특히 EGA, CGA, MDA, 허큘레스 등의 다양한 화면모드를 하나의 비디오보드로 지원이 가능하도록 설계되었다. 한편 한글 코드가 같은 한글 응용소프트웨어는 모두 사용이 가능하며, 2바이트 완성형 한글코드 및 IBM 2바이트 한글코드의 지원이 가능하다. 사용가능한 OS로는 MS-DOS, OS/2, 유닉스 머지/386, 제닉스 IRMX 등이며 각종 OA용은 물론 CAD, CAM, 과학기술계산, 공정제어 시스템 등에 활용된다.

한편 현재의 32비트 PC의 시장가는 600~800만원선으로 공식화 되어 있다. 판매가가 제자리를 잡지 못하고 각 업체들마다 가격의 차이가 나는데 각사의 출하가격은 다음과 같다.

삼보컴퓨터 TG-386이 685만원선, 쌍용컴퓨터의 드래곤 386이 660만원, 고려시스템의 고려 386이 690만원선, 대우통신의 PRO-5000이 640만원선, 엘렉스컴퓨터 MacII가 776만원선, 동양시스템산업(주) OPC-386이 550만원선으로 출하되고 있다.

## 〈용어해설〉

**●입·출력 채널(Input-output Channel)**

컴퓨터의 입·출력장치의 전송속도는 컴퓨터 본체의 연산처리 속도보다 훨씬 느리기 때문에 입·출력장치의 제어만을 행하는 장치가 필요하다.
이에 따라 중앙처리장치의 지시에 따라서 입출력장치와 메모리 사이에서는 중앙처리장치의 동작과 독립해서 데이터의 전송을 입·출력 채널이라는 장치에 맡기게 하는 방식을 채택하고 있다. 이와 같이 하면 컴퓨터는 중앙 처리 장치(CPU)에서의 데이터 처리와 입·출력 기기 조작을 병행할 수 있게 되어 있다.

**●바이트(Byte)**

컴퓨터에서 취급하는 정보의 단위를 말한다. IBM사에서 1바이트를 8비트로 정하고부터 다른 컴퓨터메이커에서도 이것을 사용하고 있다. 특별한 주석이 없는 한 1바이트는 8비트라고 해석해도 좋다. 주기억 장치의 어드레스 방식이 바이트단위로 되어 있거나 CPU액세스 단위가 바이트단위로 되어 있는데 이런 컴퓨터를 바이트머신이라고 한다.

**●인터럽트(Interrupt)**

어떤 프로그램의 실행을 일시 중단하여 나중에 다시 실행시킬 수 있도록 하고, 다른 프로그램으로 실행을 옮기는 것을 말한다. 컴퓨터는 일반적으로 몇가지의 인터럽트를 동시에 접수할 수 있으며 각 인터럽트는 우선순위가 붙여져서 이 우선순위가 높은 순서대로 처리된다.
한편, 중요한 프로그램을 실행하고 있을 때는 특정의 인터럽트만을 허용하도록 인터럽트를 금지하는 비트를 내세워 마스크를 걸어 두는 경우가 있다.

**●모듈(Module)**

메모리나 각종 장치·회로·프로그램 등을 어떤 기능별로 분할하여 증설이나 개정이 용이하게 이루어지도록 한 블록단위를 모듈이라고 한다.
예를 들면 8k 메모리가 하나의 모듈로 되어 있으면 8k, 16k, 24k…로 증설할 수 있다. 따라서 기능별로 묶여진 각 부분(모듈)의 조합으로 하나의 시스템을 구성하는 것을 모듈화라고 한다.

**●바이너리(Binary)**

2진수라고도 하며 2진수는 2를 기수로 하는 수의 체계인데 컴퓨터에서는 2진법으로 정보를 나타내어 처리하고 있다. 2진수 표현은 "1"과 "0"과의 조합이기 때문에 데이터 표현이 길어져서 알기 어렵다. 이러한 이유로 8진수나 16진수로 나타내는 방법이 편리하다.
2진수의 최소 표현 단위를 비트(Bit)라 하고, "0"과 "1" 또는 "하이"나 "로우"로 표시하고 있다.

**●패치(Patch)**

프로그램이나 데이터에 오류가 있는 경우 그 원래의 프로그램까지 거슬러 올라가지 않고 임시적인 수단으로 수정하는 것을 가르키며, •오브젝트 카드의 수정 •프로그램의 로드시에 있어서 수정 카드에 의한 수정 •실행시에 있어서 콘솔에서 행하는 수정 등 세가지 방법이 있다.

**●메모리(Memory)**

계산이나 제어에 필요한 프로그램이나 데이터를 저장하는 장소 또는 장치를 말한다. 마이크로컴퓨터에서는 롬이나 램에 프로그램이나 데이터가 저장된다. 메모리는 2진수를 받아들이거나 유지하고, 코어, 디스크, 드럼, 반도체 메모리 등이 있다.

**●오버플로우(Overflow)**

컴퓨터내에서 연산결과가 주어진 허용자리수를 넘어서는 경우를 말한다. 예를 들면, 레지스터가 취급할 수 있는 수의 범위를 입력한 계산결과를 표시하는 상태나 프로그램의 크기가 기억용량을 넘는 상태를 말한다.

**●레지스터(Register)**

플립플롭 회로를 7개 접속시켜 데이터의 산술연산·논리연산·데이터전송 등을 용이하게 하기 위해 사용되는 일시 기억용의 회로를 말한다. 레지스터의 종류에는 어큐뮬레이터·명령 레지스터·I/O 레지스터 등이 있다.

# 제6회 마이크로로로보트 경연대회를 보고

김형중/서울대 제어계측공학과 박사과정

## 환호하는 관중

지난 10월 29일, 서울대학교 교정 전체가 정적에 묻혀 초가을 상큼함이 더욱 돋보였다. 오가는 이의 발걸음이 무료함을 덜해주었다는 표현이 적절할지도 모를 만큼 온통 수채화같이 차분한 분위기가 감돌았다. 전날 축제가 끝난 탓이었을까.

그러나 본관 앞 노란색 건물, 문화관 소극장은 대조적이었다. 많은 젊은이들이 무대에 시선을 집중시켰다. 간혹 꼬마와 그들의 부모인듯한 중년의 부인과 신사의 모습도 보였다. 그들은 숨을 죽였다. 가끔 안타까워하기도 했다. 킬킬대기도 했다. 소극장에서 연극이나 영화를 관람하는 것이 아니었다. 그러면서도 그들은 매우 진지했고 무대위에서 일어나는 모든 사건에 호흡을 같이 했다.

해마다 이맘때면 꼭 열리는 전국 마이크로 로보트 경연대회가 열리고 있었다. 이날도 예선을 통과한 열두팀이 본선을 치루게 된 것이다.

처음 경연에 나선 로보트는 "바람돌이". 현재 서울대학 제어계측공학과 석사과정에 재학중인 이준화씨의 작품이었다. 이 로보트는 이 대회에 세번째 참가하는 것이라는 소개가 있었다. 특히 이전의 두번은 다 우승후보였으나 결과는 모두 입상권 밖으로 밀려났을 뿐이었다. 작년 대회가 끝난 후에는 지도교수로부터 꾸지람까지 들은 터였다. 로보트를 손질하느라 연구를 다소 소홀히 한 터였기 때문이다.

출발지점을 떠난 몇분간은 그래도 이곳 저곳 잘 돌아다녔다. 그러다 벽을 들이받고 말았다. 그 버릇이 또 도진 것이다. 그보다는 프로그램 오류를 완전히 제거하지 못해 로보트가 상황을 잘못 판단했다고 해야 더 옳을 것이다. 아뭏든 주어진 시간을 거의 다 소진한 후에 아쉬움만 남긴채 무대에서 내려섰다.

그 이후는 아주 한심한 상황의 연속이었다. 서너 블록이나 진행한 뒤 로보트가 정신을 잃었는지 뱅글뱅글 돌지를 않나 벽을 부수려는듯 자꾸 들이받지를 않나 대회의 권위를 의심할 만큼 맥빠지게 만들었다.

그러다 작년대회 우승자 전윤호씨가 등단하자 갑자기 장내가 활기에 넘치기 시작했다. 그를 기억하는 많은 관중이 힘찬 박수를 보냈다. 로보트 경연대회가 배출한 최고의 스타. "스피디" 시리즈의 주인공. 매스컴이 주목하는 젊은 영웅.

그의 로보트 "스피디 3호"가 날렵한 모습으로 출발했다. 자신에게 부여된 명성에 걸맞게 "스피디 3호"는 매끄럽게 미로를 누비고 다녔다. 관중들은 침을 삼켰다. 온통 장내는 고요함 그 자체로 가득했다.

관중의 기대에 부응하기로 작정이라도 한 것처럼 로보트는 목표지점에 돌입했다. 그것도 일차주행에 성공한 것이었다. 순간 장내는 온통 환성과 박수의 물결로 넘쳤다. 누가 먼저랄 것도 없이 관중들은 모두 기립해 있었다. 무한히 즐거운 일을 만난듯이, 그리고 그것이 자신의 일인 것처럼 한동안 그렇게 서 있을 태세였다.

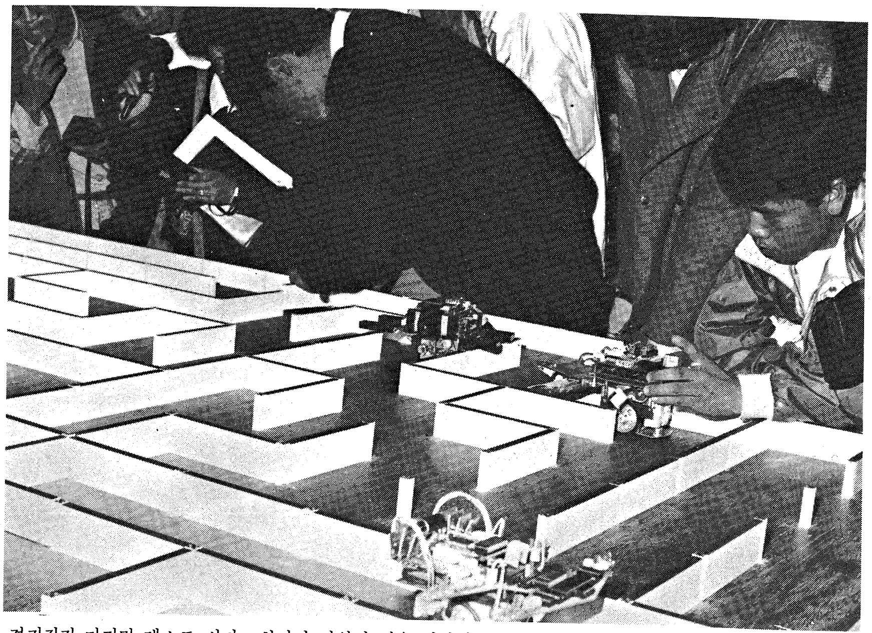
그런데 곧 그 환성과 박수가 사그러들었다. 로보트가 부지런히 출발지점으로 돌아 나오기 시작했기 때문이다. 출발지점에 도달했을 때의 공식 계시 기록이 1분 49초 98. 이어 이차주행이 시작되었다. 2차 주행기록은 22초 93. 신기록이 작성된 것이다. 그가 작년 대회에서 세운 최고기록 38초 25를 갱신한 쾌거가 이룩되었다. 참고로 입상자 로보트의 기술적 제원과 경연 기록을 표 1과 표 2에 보인다.

## 기록은 깨기 위해 있는 것

세계 최고봉인 에베르스트가 1953년 5월 29일 뉴질랜드의 등산가 힐라리와 셀파 텐진에 의해 정복되었다. 최고봉 그 자체가 정복되어야 할 대상이자 표적이 되는 것이다.

기네스 북은 최고의 기록을 세우고자 노력하는 인간의 무한한 욕망 가운데 하나인 도전이라는 욕구의 산물이다. 1956년 창간되어 지금은 세계에서 가장 인기있는 책이 되었을 뿐만 아니라 영국의 각 도서관에서 가장 잘 분실되는 책이라는 기록도 지니고 있어 유명하다. 심오한 학문의 영역에서부터 기록이라고 부르기에는 다소 어처구니 없는 부문에 이르기까지 수천 항목이 있는데 이 책에 이름을 남기기 위해 위험을 무릅쓰다 사고를 일으키는 일이 빈발해 편집인이 경고를 받기도 할 정도이다.

올림픽도 예외는 아니다. 기록은 깨기 위해 존재한다는 것의 산 증거가 바로 올림픽 경기장에서 실증된다. 모든 국민의 자존심을 고양시켜준 서울 올림



경기직전 마지막 테스트 시간—참가자 전원이 매우 긴장해 있다.

픽에서 열두개의 금메달을 획득한 것도 한국 신기록이었다. 벤 존슨이 약물복용으로 치욕적인 경기인 생활의 한 획을 그은 것도 바로 기록 때문이다.

기록은 부정적인 면도 지니지만 그러나 대부분은 그 속성 때문에 스타의 탄생을 약속한다. 칼 루이스, 마이클 잭슨,…. 그것은 본질에 앞선다는 철학적 명제의 적용 대상이기도 하다. 마이클 잭슨의 "드릴러"가 빌보드 챠트 정상을 가장 오래 차지했다는 사실 그 자체가 그 음악의 가치를 어느 정도는 높여주었기 때문이다.

그날 전윤호씨의 기록은 참으로 값진 것이었지만 그것은 더 좋은 기록의 출현을 위한 서막에 불과했다. 3분 규정을 적용받는 선수들의 경연 순서가 이르렀다. 작년대회 우수상 수상자인 박남규씨가 "터보 4호"를 들고 등단했다. 그에게 부여된 시간은 5분. 그의 로보트는 마지막 남은 시간을 아껴 조심스럽게 움직였다. 그의 로보트도 제한시간내에 무사히 목표지점에 골인했다. 다시 장내는 환성과 박수의 도가니가 되었다. 그러나 그의 로보트가 출발지점에 돌아오다 벽을 들이받아 기록갱신도 한낱 환상에 불과한 것처럼 보였다.

다시 그의 로보트가 이차주행에 들어 가자 "터보 4호"는 17초 99라는 경이적인 기록을 남겼다. 더욱 더 감동적인 것은 아주 고난도의 기술인 스무드턴(Smooth Turn)을 선보인 것이다. 마치 코마네치가 여자 체조에서 경이적인 동작을 선보이듯이 미로를 갈랐다. 코너를 도는 "터보 4호"의 모습은 예술 그 자체였다. 이번 로보트 경연대회의 백미이자 대미를 화려하게 장식한 것이다.

비록 이차주행에서는 국내 최고기록을 세웠지만 박남규씨는 또 우수상에 만족해야 했다. 현재 서울대학교 제어계측공학과 석사과정 1학년에 재학중이며 로보트공학을 연구하는 그는 개정된 경연대회 규정의 희생타가 된 것이다.

전윤호씨는 미로시간 1분 49초 98을 사용한 상태에서 이차주행시간이 22초 93이었고 제작자의 도움 없이 완주했기 때문에 10초를 경감해주는 규정을 적용받게 되었다.

그래서 그의 경연시간은 다음 계산

〔표 1〕 입상자 로보트의 제원

| 제 작 자 | 전윤호 | 박남규 | 이준화 | 김철균 |
|---|---|---|---|---|
| 로보트 이름 | 스피디 3 | 터보 4 | 바람돌이 | 하이퍼 |
| 크기(mm) | 248×190×130 | 200×200×100 | 200×200×100 | |
| 주연산장치/클럭(MHz) | V40/8 | Z80A와 8031/4 | MC6809/2 | |
| 롬/램 (단위 : K바이트) | 8/32 | 32/8 | 8/8 | |
| 개발언어 | 어셈블리 | 파스칼, 어셈블리 | 어셈블리 | |
| 개발장치 | IBM-PC | Apple | IBM-PC | |
| 모 터 | 스텝모터 2개 | 스텝모터 2개 | 스텝모터 2개 | |
| 센서방식 | 전방 : 초음파 1개<br>측방 : 적외선 32개 | 전방 : 초음파 1개<br>측방 : 광센서 12개 | 전방 : 투과 1개<br>측방 : 반사 10개 | |
| 전 원 | 1.2V 0.5AH 12개 | 1.2V 0.5AH 10개 | 1.2V 1.8AH 6개 | |

〔표 2〕 입상자의 경기기록

| 참가팀 대표 | 로보트 이름 | 미로시간 | 주행시간 | 경연시간 | 등위 |
|---|---|---|---|---|---|
| 전 윤 호 | SPEEDY 3 | 1 : 50.10 | 22.93초$^{-10}$ | 16.60초 | 1 |
| 박 남 규 | TURBO 4 | 16 : 04.95 | 17.99초 | 50.16초 | 2 |
| 이 준 화 | 바람돌이 | 완주하지 못함 | | | 3 |
| 김 철 균 | HYPER | 완주하지 못함 | | | 4 |

$$\frac{109.98}{30}+22.93-10$$

의 결과 16초 60이 된다. 그러나 박남규씨는 이차주행에 들어가기 전에 이미 규정시간을 약간 초과한 16분 4초 95를 사용했고 이차주행에서 17초 99를 기록했지만 제작자 도움을 받아(제작자가 손으로 로보트를 들어) 출발지점에 돌아왔기 때문에 그의 경연시간은

$$\frac{964.95}{30}+17.99=50.16$$

으로 50초 16이 된다. 미로시간이 얼마나 치명적인지를 실감하게 되는 것이다. 그것이 승부 세계의 냉혹한 당면이기도 하다. 역대 대회 입상자 기록은 **표 3**과 같다.

## 대회 낙수

10월 15일 제어계측공학과 강의실에서는 예선대회가 진행되고 있었다. 이 날 경기를 지켜본 많은 관중은 홍익대 전기제어공학과 4학년 정용주, 조종일씨의 "K509호"의 움직이는 모습을 보고 박장대소했다. 그 로보트는 한 블록을 진행한 다음 마치 좌우를 둘러보듯 90°를 회전한 다음 다시 원래 방향으로 되돌려 다시 한 블록을 진행하기를 반복했다. 그것은 저능아가 이곳 저곳 기웃거리는 모습과 흡사했다.

관중의 웃음은 제작자들에게 큰 충격을 주었음을 틀림없다. 그들은 이차예선에 참석하지 않았다. 그렇지만 그 로보트는 다른 로보트에 비해 훨씬 많은 블록을 전진했기 때문에 본선에도 진출할 수 있었다.

이 로보트는 앞면에만 센서를 부착했던 것이다. 그러니 옆면에 벽이 있는지 알아보기 어렵게 된 것이다. 그래서 어쩔 수 없이 90°를 회전해서 앞벽 센서로 옆벽의 유무를 확인해야만 했던 것이다. 콜롬보 형사의 추리력으로 정황을 살피건데 그것은 분명히 새로운 시도임이 분명하다. 다만 그렇게 한 동기를 이해할 수 없다는 것이 문제이지만.

또 하나 기억에 새로운 로보트는 전북대 전자공학과 3학년의 김철균씨 로보트 "하이퍼"이다. 이 로보트는 센서라고는 온통 초음파 센서만 사용해서 완성된 것이 특징이다. 다른 로보트들은 보통 빛을 이용한 센서와 초음파를 이용하는 센서 두가지를 혼용한다. 그래서 자원봉사자들은 이 로보트를 시각장애자 2등급 로보트라 부르기도 했었다.

초음파센서는 위치보정을 완벽에 가깝게 한다. 그래서 충분한 시간만 주면

〔표 3〕 역대대회 입상자 기록

| 대 회 | 로보트 이름 | 제 작 자(당시 소속) | 입상내용 | 기 록 |
|---|---|---|---|---|
| 1회 1983. 11. 4 | SECRET | 구본호, 김동식, 박재현, 유견아(이상 서울대 제어계측 2) | 우 수 상 | 3분15초 |
| | ROPET | 박영준, 최재영, 허종성(이상 서울대 제어계측 4) | 장 려 상 | 완주하지 못함 |
| 2회 1984. 10. 27～28 | 골목대장 | 이석민, 이세형, 이왕헌(이상 서울대 제어계측 4) | 최우수상 | 1분50초 |
| | SKYLINE | 이승찬, 이용훈(이상 서울대 제어계측 2), 이우빈(서울대 전자 2) | 우 수 상 | 3분50초 |
| | 뚝딱뚝딱 | 김기회(서울대 전자 2), 남창석(수원대 전산 1) | 장 려 상 | 완주하지 못함 |
| | TRON-Ⅰ | 주석화, 김 준, 김문종(이상 Tron Computer System) | 준장려상 | 완주하지 못함 |
| 3회 1985. 7. 6 | 고주 2호 | 이 혁, 김용훈, 장덕현(이상 서울대 전자 4) | 최우수상 | 1분13초7 |
| | 뚝딱반짝Ⅱ | 김기회(서울대 전자 3), 남창석(수원대 전산 2) | 장 려 상 | 1분57초8 |
| | TRON-Ⅱ | 윤형삼(Tron Computer System) | 감 투 상 | 완주하지 못함 |
| 4회 1986. 11. 25 | 뚝딱반짝Ⅲ | 김기회(서울대 전자 4), 남창석(수원대 전산 3), 임경래(인하대 전자 4) | 장 려 상 | 완주하지 못함 |
| | 터보 2 | 박남규(서울대 제어계측 3) | 장 려 상 | 완주하지 못함 |
| | 한양 4 | 권순돈, 김형철(이상 한양대 전자 3) | 장 려 상 | 완주하지 못함 |
| | 아이큐 100 | 홍관유, 윤대희, 박경호, 변경환(이상 동양공전 전자공학과) | 장 려 상 | 완주하지 못함 |
| 5회 1987. 10. 31 | 스피디 2 | 전윤호(서울대 제어계측 3) | 최우수상 | 38초25 |
| | 터보 3 | 박남규(서울대 제어계측 4) | 우 수 상 | 56초27 |
| | 메이비 1 | 김성희, 박호용(이상 서울대 전기 4) | 준우수상 | 1분11초17 |
| | KK 2 | 권순돈, 김형철(이상 한양대 전자 4) | 장 려 상 | 1분32초68 |
| | IQ101 | 변경환, 장병국, 김종필(이상 동양공전 전자 2), 최석열(통신 2), 김형동(전자 1) | 장 려 상 | 1분38초59 |
| 6회 1988. 10. 29 | 스피디 3 | 전윤호(서울대 제어계측 4) | 최우수상 | 22초93 |
| | 터보 4 | 박남규(서울대 제어계측 석사 1) | 우 수 상 | 17초99 |
| | 바람돌이 | 이준화(서울대 제어계측 석사 2) | 장 려 상 | 완주하지 못함 |
| | 하이퍼 | 김철균(전북대 전자 2) | 장 려 상 | 완주하지 못함 |

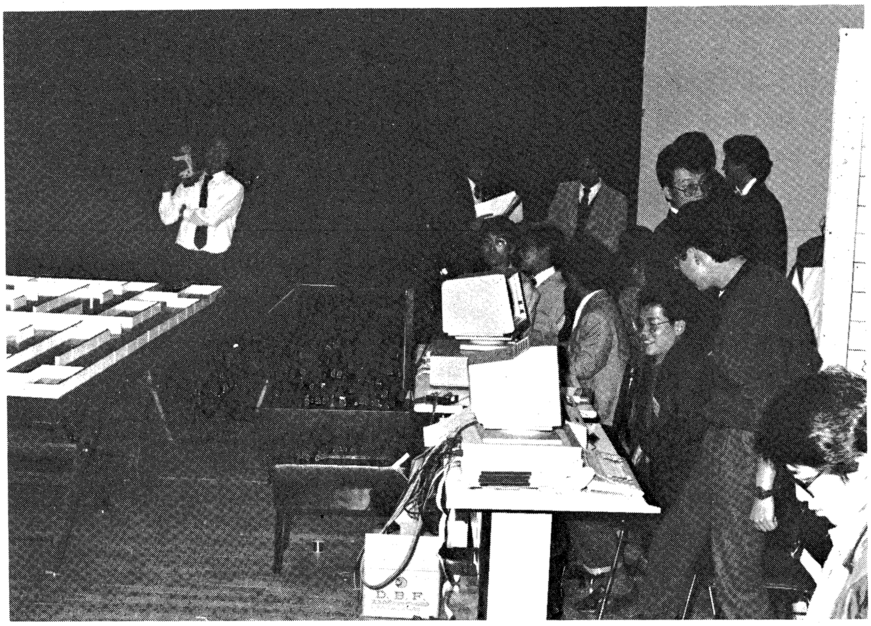
주1) 가운데 찾기 경기기록만 수록함
주2) 이 기록은 결선대회 최단시간만으로 작성된 것임

완주할 수 있을 것처럼 보였다. 그러나 실제 미로에서 충분히 테스트를 해보지 못한 탓으로 완주에는 실패했다.

이 로보트는 사진기자들의 플래쉬 세례를 받은 유일한 로보트가 되었다(그것이 나중에 김원찬 심사위원장 직무대리의 심사소감에서 수상자 선정의 한 이유가 되었음이 밝혀졌다). 원래 로보트가 주행하고 있는 동안에는 철저하게 사진기 플래쉬 사용이 금지된다. 빛을 감지하는 센서에 의존하는 로보트에게 플래쉬의 섬광은 치명적인 사이비 종교 교주처럼 잘못된 길로 인도할 염려가 있기 때문이다. 그러나 초음파에만 의존한다면 듣지 못하는 이에게 불경 읽어주는 식이니 전혀 영향을 받지 않는다.

이번 대회에서 가장 안타까웠던 일은 서울대 로보트 "종횡무진"의 제작자 방영봉씨가 본선에 출전하지 못하고 기권으로 처리된 일이다. 현재 서울대 기계공학과 4학년에 재학중인 그는 이 대회에 올해로 네번째 참석한다. 그런데 한번도 입상한 적이 없는 불운을 감수해야 했다. 이번 대회에서는 그것을 설욕할 것처럼 보였다. 예선에서 그의 로보트는 간단히 완주에 성공했던 것이다. 조금 느리기는 했지만 기계공학도가 전자공학적인 지식을 홀로 습득해 이처럼 완벽하게 움직이는 로보트를 제작했다는 것이 믿어지지 않을 정도였다. 실제로 그의 로보트는 훌륭했고 특히 기계적인 부분은 타의 추종을 불허할 정도로 완벽해 M48 탱크의 완충장치 정도는 되어 보였다.

그런데 일이 터진 것이었다. 그는 본선 참가번호 1번이었다. 선수입장식에서 사회자가 그의 이름을 부르자 탄호이저의 행진곡이 우렁차게 울리기 시작했는데 그는 나타나지 않은 것이다. 대회가 한창 진행중일 때 전갈이 왔다. 다소 늦어지겠다고. 실제 그가 나타난 것은 대회가 끝난 후 자원봉사자들이 미로를 완전히 해체한 다음이었다. 아침에 갑자기 센서가 고장났다는 것이었다. 또 하나의 장애자 로보트가 탄생한 것이다.



자체제작한 계시장비로 기록계시에 열중인 기록계시 요원들.

## 장난감인가

마이크로 로보트는 장난감인가? 그럴 수도 있다. 장난감 회사에서 그것을 장난감이라고 팔면 장난감이 되는 것이다. 그러나 장난감치고는 너무 고급스럽고 정교하다.

그렇다면 그것은 로보트인가? 이것이야말로 틀림없는 사실이다. 왜냐하면 로보트 경연대회에 출전했기 때문이다. 이것은 지극히 우스꽝스러운 논리이지만 로보트가 어떠해야 한다는 규정이 없는 한 이러한 동의어 반복은 피할 수 없다.

로보트는 보통 관절로보트, 주행로보트, 보행로보트 등으로 나눌 수 있다. 관절로보트는 관절을 구부려 이동하는데 관절이 많을수록 자유도가 커지지만 제어하기 어렵게 된다. 조립공장에서 나사를 조이거나 부품을 끼우는 로보트가 이런 부류에 속한다.

주행로보트는 바퀴나 캐터필러를 달아 움직이는 로보트를 말한다. 로보트는 보통 무인로보트를 의미하므로 자동차와 구별된다. 이런 로보트는 물건을 운반하는데 주로 이용되는데 센서의 한계로 인해 자유공간을 마음대로 움직이게 하기는 어렵다. 그래서 보통 바닥에 형광도료나 자성도료를 칠한 다음 그 위를 이동하게 하는 방식을 쓴다. 그것에 비하면 주행로보트에 속하는 마이크로 로보트는 대단히 정교한 것임을 알 수 있다.

보행로보트는 관절을 움직여 이동하는 것은 관절로보트와 다를 것이 없다. 그러나 보행로보트는 지면에서 한 부분을 들어 올려 계단을 오르는 동작을 할 수 있어야 한다는 점이 다르다. 따라서 균형과 무게중심의 유지가 생명이다. 로보캅이 이런 유형에 속한다.

그렇다면 왜 장황하게 로보트의 종류를 설명했는가? 보통 장난감이라면 상당히 깔보는 듯한 여운이 남는다. 기술적으로나 상업적으로 가치가 없다는 투가 배어있기 때문에 결코 그렇지 않다는 것을 강변하고 싶어서 이야기를 잠시 길에서 벗어나게 한 것이다. 주어

진 미로에서 이처럼 치밀하게 움직이도록 제작할 수 있는 기술이라면 보다 덩치가 크고 세련되어 보이는 로보트도 제작이 가능한 것이다.

상업성에서라면 그 활용도는 무궁무진하다고 말할 수 있다. 우선 장난감으로 제작해 팔 수 있다. 원자로나 화공약품 탱크와 같이 위험한 환경에서 벽면의 균열 등을 검사하는 데도 마이크로 로보트를 쓸 수 있다. 파이프의 막힌 곳을 찾아내게 할 수도 있다. 무인진공청소기나 무인자동차의 제작에 활용하지 말라는 법도 없다.

알렉산더 그레엄 벨이 전화기를 발명했을 때 그것이 정보화된 사회 출현의 기폭제가 되리라고 예상하지는 못했을 것이다. 애플컴퓨터가 등장했을 때도 그것이 몰고올 변혁의 흐름을 읽지는 못했을 것이다. 이제는 누구나 값싼 자동차 한대 가격이면 애플컴퓨터가 출현할 당시의 중형컴퓨터에 비교되는 개인용 컴퓨터를 살 수 있게 된 것이다.

그래서 앨빈 토플러와 같은 위대한 미래학자의 예측도 어느 정도 대중화된 연후에야 그것이 몰고올 가능성을 제시하는데 그치고 만다. 인간의 상상력이란 자신이 속한 현실을 초월할 수 없다는 진리가 실감나는 것이다. 낙엽이 지는 것을 보고 가을이 왔음을 안다는 시인의 영탄이 바로 여기에 해당한다

로보트, 마이크로 로보트가 몰고올 충격은 결코 작지 않을 것이다. 그러나 그것을 장난감이라고 업신여기는 동안에는 그저 장난감에 불과하게 될 뿐이다. 사물의 효용이 어떻게 극대화되는가? "성자가 된 청소부"(바바 하리 다스 지음, 류시화 옮김)의 한 페이지를 보자. 거기서 돼지의 효과적인 이용방법을 하나 살피게 된다.

자반의 어머니는 자반이 얻어 오는 음식 중에서 먹고 남은 것으로 돼지를 치기 시작했다. 돼지들은 거리를 청소하는데 도움이 되었다. 아침 일찍 자반의 어머니가 돼지막 문을 열어 주면 돼지들은 거리로 달려 나가 잔뜩 쌓인 쓰레기들을 먹어 치웠다. 이것은 자반의 할 일이 그만큼 덜어진다는 것을 의미했다. 이따금 하수구가 막히면 돼지들이 하수구 속으로 들어가 구멍을 막고 있는 쓰레기 더미를 밀어내 주기도 했다.

마이크로로보트가 몰고 올 충격은 결코 작지 않을 것이다.
그러나 그것을 장난감이라고 업신여기는 동안에는
그저 장난감에 불과하게 될 뿐이다.

## 자존심을 지켜주다

제4회 대회가 끝난 후 일본의 "노리코"라는 로보트가 우리의 자존심을 여지없이 짓밟아버린 일이 있었다. 공식적으로 유일하게 완주한 팀이 없었던 이 대회를 비웃기라도 하는 양 "노리코"는 유연한 몸짓으로 가볍게 골인했고 그것도 당시로서는 상상할 수 없는 18초대에 주파해버린 것이다.

그날 관중은 "노리코"의 모습이라도 보지 못했다면 아마 다시 로보트 경연대회에 참관한다면 손에 장을 지지겠다고 결심했을 법했다. 오전 열시부터 오후 네시까지 출전한 로보트들이 연출한 내용은 대동소이했다. 몇블록 간 다음 부딪히고…. 그러면 다시 들어내 나사를 풀고 저항을 바꿔보고 센서에 이상이 있는지 이리저리 움직여보고…. 그리고 다시 출발시키면 또 조금 가다가 멈추고….

그래서 출전선수들도 절치부심했다. 타도 "노리코"를 외치면서 제5회 대회를 열었는데 38초대에 들어서게 되어 앞이 조금 환해 보이기 시작했다. 그러나 일부 관측통들은 그것이 한계일 것이라고 믿었다. 기술은 역시 일본이라는 고정관념 때문이었을게다. 그해 전윤호씨가 명색이 국제대회라지만(우리들은 그렇게 부르는 것이 관례화되어 있다. 그러면 사람들은 대단히 많은 나라의 선수들이 엄격한 예선을 거쳐 출전했으며 거기서 입상하기가 하늘의 별따기 만큼이나 힘들 것이라는 선입관을 갖게 만드는 마력이 있기 때문에) 캐나다, 미국, 한국의 선수가 모여 조촐하게 한번 시합해본 결과 전체 3위를 기록하고 돌아왔다. 학생부에서는 1위를 차지했고.

아뭏든 그것은 대단한 기록이었다. 이미 그 전에 국내 마이크로 로보트 경연대회에서 입상한 두 팀이 일본에서 열린 국제대회에서는 완주조차 못한 경험이 있기 때문이다. 그래서 전윤호씨의 입상은 더욱 값진 것이다.

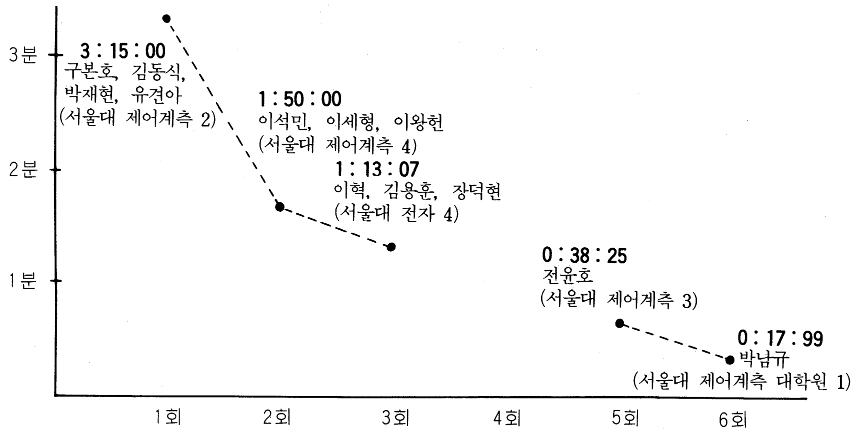
올해 대회에서 세워진 기록 17초 99와 22초 93은 우리의 자존심을 세워주고도 남음이 있다. 이제 세계대회에 나선다고 해도 결코 부끄럽지 않은 결과를 얻기에 충분한 기록이기 때문이다. 역대 대회의 최고기록 추이를 보면 다음 **그림 1**과 같다.

## 프로근성

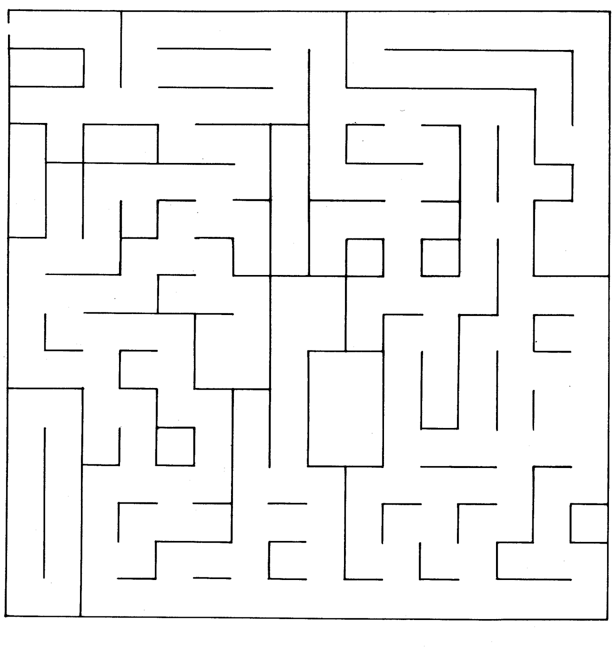
전윤호씨가 서울대학교 제어계측공학과 2학년에 재학중일 때 만든 최초의 로보트는 제어가 불가능한, 그래서 가미가제 특공대의 비행기처럼 난폭하게 돌진하다 무모하게 최후를 마치는 로보트의 전형이었다. 그러나 로보트는 전자회로만이 완벽해서 움직이는 것은 아니라는 신념에 따라 안정된 기계적 구성에 주력했다. 소프트웨어 작성에 있어서의 효율성도 높였다. 매번 프로그램을 완성한 다음 롬에 구워넣어 동작시켜보고 오류가 있으면 다시 작성하는 비효율적인 방법을 지양했다.

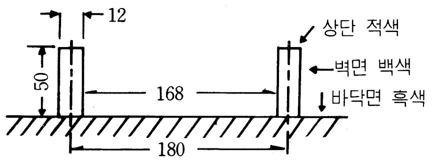
박남규씨도 마찬가지이다. 그는 대회에서 운이 따르지 않았지만 새로운 기

〔그림 1〕 대회 최고기록 추이

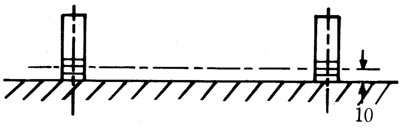



〔그림 2〕 제6회 마이크로로보트 경연대회 본선 미로

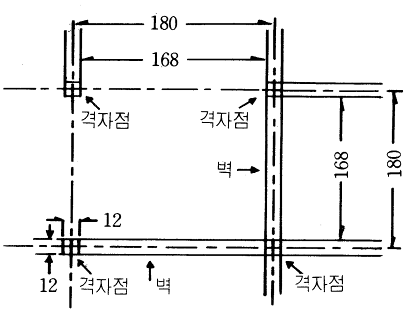





미로의 단면도(단위 ㎜)




센서의 위치(단위 ㎜)




미로의 평면도 한 블럭(단위 ㎜)

술의 창안자이다. 그가 창안한 최고의 기술은 스무드턴이다. 보통의 로보트는 회전을 해야 되면 한쪽 바퀴는 정지시켜 회전축으로 삼고 나머지 한쪽 바퀴만 90° 돌린다. 그러면 직선코스를 달리는 동안 생긴 관성을 일단 멈춤으로 해서 제거하기 때문에 회전에만 주력하면 된다.

그러나 스무드턴을 하기 위해서는 정지하지 않기 때문에 자신이 어떻게 움직이게 될지를 정확히 예측하지 못하면 안된다. 그것은 대단히 어려운 일이며 많은 계산시간을 요한다. 섬세하고 예리한 관찰력이 필요한데 일단 완성시키기만 하면 환상적인 움직임과 고속주행이 가능해진다. 보통 정지해서 회전하는 것을 고정표적에 사격하는 것과 같다면 스무드턴은 이동표적에 사격하는 것과 같다.

이들은 내년에도 또 이 대회에 출전할 것이다. 또 선의의 경쟁을 선보일 것이다. 이제 장롱 속에 넣어두었다가 내년에는 "먼지만 털고" 다시 출전하겠다고 농담하기는 했지만 그들은 다시 프로근성을 발휘할 것이다. 더 좋은 성적을 내기 위해, 더 빨리 달리기 위해, 더 안정되게 만들기 위해서. 그들이 쌓은 경험과 체계적인 기술, 그리고 자료는 로보트 제작의 필수적인 정보가 될 것이며 아마츄어들이 본받아야 할 모범인 것이다. 1초, 아니 0.1초가 중요한 승부의 세계에 들어서고 싶다면 말이다.

이들은 미로의 모양에도 세심한 배려를 쏟는다. 최단경로가 50블록인데 20번 회전해야 하는 코스와 70블록을 가야 하는데 10번 회전해야 된다면 어느 경우를 선택해야 될 것인가. 얼핏 생각하면 최단경로가 최선의 선택인 것처럼 보인다. 실제로 그럴 수도 있다. 그러나 그것은 로보트의 성능과 밀접한 관계가 있음을 간과해서는 안된다. 회전에 강하다면 직선코스에서도 강하다는 뜻이 되는 것이 경험상으로 반드시는 아니지만 상당히 정확한 표현이 될 수

있다. 이런 로보트는 최단코스를 달리는 것이 유리하다. 직선코스에서는 아주 자신이 있는데 회전에 약하다면 원칙을 세워야 된다. 회전대 직진의 비율이 어느 정도일 때 어떤 경로를 택할 것인지는 자신만의 비밀이다. 프로의 세계에서 이것도 중요한 문제이다.

## 두 다리만 튼튼하다면

"천리길도 한걸음부터"라는 속담이 말해주듯이 멀리 가고 싶다면 한걸음 한걸음을 다 소중히 생각해야 한다. 한 블록도 정확히 움직이지 못하는데 50블록 이상 가리라 기대하는 것은 어리석다.

그러나 해마다 거의 대부분의 로보트는 이런 상태에 머물고 있다. 밤을 지새우고 청계천으로 구로동으로 부품을 구하기 위해 발바닥이 닳도록 뛰어다닐 때는 그래도 다 로보트다운 로보트를 만들겠다는 일념에 사로잡혔겠지만 만들어진 것은 불안정하기 그지 없다.

그 원인의 대부분은 실물 크기의 미로가 없기 때문이리라. 자유공간에서 잘 움직이는 정도는 실패와 고무줄 그리고 연필로 만든 1960년대에 집에서 장난삼아 만들어보았던 장난감으로도 가능하다. 자동차가 차도를 벗어나 인도로 뛰어든다면 누가 그것을 자동차라고 부르겠는가?

규정된 미로에서 정확히 움직이도록 하는 것이 기술의 시초이다.

또 하드웨어가 완성되고 나면 바로 로보트가 된다고 믿는 것이 실패의 지름길이다. 하드웨어와 소프트웨어는 불가분의 관계에 있는데 작품을 완성하는데 30%의 시간과 노력을 전자에 쏟았다면 후자에 나머지 70%를 쏟아야 원하는 동작을 한다는 것이 전문가들의 통설이다. 로보캅이라는 영화에서 최초로 만들어진 로보트를 생각해보라. 하드웨어는 완성되었는데 소프트웨어가 잘못되어 적이건 아군이건 표적이 되면 총을 난사하게 되는 비극을 유발하게 되었다.

하드웨어를 제작할 때에도 기판의 30% 정도는 비워두어야 한다. 처음부터 완벽하게 회로를 설계할 수 있는 사람은 없다. 설계도대로 제작하다보면 잡음을 제거하는 회로가 추가되어야 하기도 하며 처음 설계가 잘못되어 변경하기도 해야 하는 상황이 언제나 생긴다. 그런데 처음부터 기판에 여유를 두지 않으면 나중에는 기판 전체를 새로 제작해야 하는 일이 발생한다. 비상금이 있으면 여유가 있듯이, 욕심을 버리고 마음 한 구석을 비워두면 행복하듯이.

또 다시 한번 더 정보교환의 중요성을 강조하고 싶다. 혼자 열심히 연구하는 것은 나무랄데 없지만 남이 다 해본 것도 모르고 젊음을 불태우는 것은 낭

●미니 인터뷰/**최우수상 수상자**●



## 새로운 기술들이 계속해서 개발되는 계기가 되었으면

**전윤호**/서울대 제어계측 4

"남규형이 더 빨랐는데…. 제가 운이 좋았던 것 같아요."라고 겸손한 웃음을 지어 보이는 제6회 마이크로로보트 경연대회 최우수상 수상자인 전윤호씨.

이번에 우승한 그의 스피디-3호는 루프 테스트 기능과 초음파 센서, 적외선 센서 등을 부착, 정확하게 벽을 감지하여 경기시간을 단축하여 많은 사람들의 인기를 끌었다. 그러나 무엇보다도 속도 단축의 원인은 여러가지 하드웨어적인 장비보다는 소프트웨어의 우수성에 있다고.

이번 대회로 세번째 참가하는 전윤호씨는 고등학교 시절부터 마이크로로보트에 관심을 가졌었고, 경연대회에 참가하고자 실제로 여러가지 관련 자료들을 모으며, 충분히 기술적인 지식을 쌓은 후 제작하였다고 한다. 매사에 꼼꼼하고 치밀한 성격의 소유인 그의 한 단면을 볼 수 있는듯. 로마는 하루아침에 이루어지지 않는다는 말이 실감이 간다.

그는 또한 마이크로로보트를 제작하면서 가장 어려웠던 점으로 시스템 노하우에 대한 지식을 든다. 대부분의 부품들이 국내에서 구하기 어려우며, 특히 관련된 자료들을 전혀 구할 수 없다는 점이 아쉽다고 한다. 그렇지만 신중하게 하나씩 해결하는 자세를 가지고 이를 해결해 나갔다고.

"마이크로로보트를 만드는 사람은 한번 시도해서 실패했다고 포기하지 말고 좀더 보완해서 다시 한번 시도하려는 의지가 있어야 합니다. 그리고 현재의 경연대회는 속도면을 중시하고 있는데 똑같은 크기의 미로가 아닌 불규칙성을 가미한 미로를 사용하는, 내용면에서 다른 경연대회가 하나 더 있었으면 합니다."라고 말하며, 다음 대회에서는 스무드턴 알고리즘을 사용하여 경기에는 참가하지 않고 시범경기만 보이고 싶다는 그의 말에서 자신의 마이크로로보트를 능가하는 새롭고 진보된 마이크로로보트가 많이 탄생하기를 바라고 있음을 알 수 있다.

지금까지 부상으로 받은 컴퓨터는 전부 어디에 사용하느냐는 기자의 질문에 그저 부끄러운듯 겸연쩍게 웃기만 한다.

비이다. 실패를 많이 경험한다는 것이 실패하지 않는 비결을 알게하지는 못한다. 값진 실패도 있지만 대부분 실패는 실패 그 자체이다. 앞선 기술 가운데 전수받을 수 있는 것은 어떻게든 자기 것으로 만드는 것이 중요하다.

몇년 전만 해도 한국이 100억달러 수출을 외쳤지만 지금은 무역흑자가 100억달러를 육박해 고민하고 있다. 올림픽에서 금메달을 여섯개 획득하면 난리라도 날 것처럼 생각했는데 그 곱절의 수확을 거두었다. 자신감에 넘쳐있는데 우리라고 못해낼 리 없다.

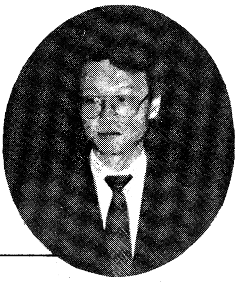
이제는 인간을 모델로 삼는 새로운 연구들이 진행되고 있다. 신경컴퓨터가 그것이고 로보트도 마찬가지이다. 무한히 새로운 것을 추구하며 전진하기를 원하는 젊은이들에게 로보트 경연대회는 좋은 무대이다. 두 다리를 튼튼하게 하면 다시 말해 한걸음이라도 소중히 생각하면서 안정되게 움직이는 로보트를 만든다면 언젠가는 골인하게 될 것이다.

열린 지평선. 그 열린 상태 때문에 오늘도 잠을 설치며 궁리하는 젊은이가 있다. 내일의 영광을 그리면서. 행동하는 젊은이가 내일을 연다. 그래서 다음 대회가 기다려진다. 전윤호씨가 연거푸 3년 연속 우승해서 부상으로 받은 IBM-PC가 세대가 될 것인지 아니면 새로운 별이 나타나 우리를 놀라게 할지 궁금해진다.

아뭏든 내년에도 로보트는 구를 것이다. 또 벽을 들이받을 것이고 킬킬댈 것이며 환호가 터지기도 할 것이다. 내년에도 어김없이 가을이 오듯이. PC

●미니 인터뷰/**우수상 수상자**●

# 더 많은 사람들이 마이크로로보트와 접할 수 있기를



**박남규**/서울대 대학원 제어계측과 1

"1차 주행시 효과적인 미로 탐색을 위해 루프 테스트에 관한 소프트웨어를 보강하지 못한 원인 때문입니다."라고 약간의 아쉬움을 담고 1등에 당선되지 못한 이유를 프로그램의 불완전으로 돌린다.

그는 실제적인 주행시간이 17초99라는 최고의 기록을 세웠지만 개정된 규정에 의해 최우수상의 영광을 뒤로 해야 했지만 이번 마이크로로보트 경연대회에서 스무드턴이라는 새로운 알고리즘을 시도함으로써 많은 사람들에게 충격을 주었다.

마우스란 주제 자체가 기계, 전기, 전자, 전산 등의 다양한 분야를 포함하고 있기 때문에 3년동안 마우스와 접하면서 많은 것을 배우게 되었고, 실제 자신의 전공에 많은 도움을 받았다고 하는 그의 말에서 철저한 마우스 애호가임을 알 수 있다.

그는 또한 내년에는 자이로스코프, CCD센서, DC모터 등 하드웨어적인 기술들이 개선되고 소프트웨어도 좀더 수정되어 정말로 새롭고, 빠르고, 정확한 마이크로로보트가 나오길 기대하며, 더 많은 사람들이 끈기있게, 창의력을 배양하면서 마이크로로보트와 접할 수 있기를 바란다고 한다.

내년의 출전여부에 관하여는 "시간이 허락된다면…"이라고 말끝을 흐리지만 그는 분명히 소프트웨어를 보강하고 모터의 힘을 크게 하여 다시 한번 도전할 것으로 믿어 의심치 않는다.

# 제6회 전국 마이크로로보트 경연대회 경기규정

**1. 미로에 관한 규정**

① 미로는 18cm×18cm의 장방형 단위구역의 16×16개로 구성되며 미로를 구성하는 벽은 높이 5cm, 두께 1.2cm이며 미로의 통로는 16.8cm이다.

② 미로의 벽 옆면은 흰색이며, 윗면은 적색이다. 미로의 바닥은 나무로 만들어져 있으며 무광택 검은 도료가 칠해져 있다.

③ 미로의 시작점은 네 모퉁이 중 하나에서 출발하며 시작 지점은 세벽으로 막혀 있다. 미로의 중앙부에는 네개의 단위 구역크기의 도착점이 있다. 도착구역에는 높이 20cm, 각변이 2.5cm인 빨간색 사각기둥이 세워져 있으며 이것은 제작자의 요청에 의하여 제거될 수 있다.

④ 각 구역의 네 귀퉁이에는 가로 세로 1.2cm, 높이 5cm의 기둥이 있다. 도착지점을 제외한 모든 기둥은 한개 이상의 벽과 연결되어 있다.

⑤ 전체 미로 크기는 5% 이내의 오차를 가지고 있으며, 미로의 바닥의 연결부위는 1mm 이내의 격차를 가지고 있다. 그리고 벽과 벽 사이의 간격은 1mm 이내이다.

**2. 마이크로 마우스에 관한 규정**

① 마이크로 마우스는 자립형이며 독립적인 에너지원을 가져야만 한다. 연소를 이용한 연료를 사용해서는 안된다.

② 마이크로 마우스의 크기는 25cm×25cm 이내이어야 하며 이 크기는 주행 중 계속 유지되어야 한다. 그러나 높이에는 제한이 없다.

③ 마이크로 마우스는 주행중 미로에 어떠한 것도 남겨서는 안된다.

④ 마이크로 마우스는 미로벽을 뛰어넘거나 기어 넘어서는 안되며 벽에 흠집을 내거나 파손을 해서는 안된다.

### 3. 경기에 관한 규정

① 마이크로 마우스의 기본적인 기능은 미로의 시작지점으로부터 종착지점을 찾아가는 것이며 이것을 주행이라 정의한다. 또 주행시 걸리는 시간을 주행시간이라 정의한다.

단, 목적지로부터 돌아오는 주행시간은 주행에 포함되지 않는다. 마이크로 마우스를 처음 동작시킨후 매 주행을 시작할 때까지 미로시간이라 정의한다. 경기 점수는 마이크로 마우스의 속력, 미로를 찾는 능력, 자립도 등을 고려하여 측정한다.

② 마이크로 마우스의 경연시간은 매 주행시간에 미로시간의 30분의 1을 더하여 계산되며 만일 제작자의 도움없이 주행을 마치게 되면 경연시간에서 10초를 감해 준다.

*경연시간=주행시간+미로시간/30−(−10초)

예) 주행시간이 30초이고 처음부터 6분이 경과했을 경우, 경연시간은 32초가 된다.

경연시간=30+(6×60)/30−10
=30+10−10=32초

③ 모든 경연시간중 가장 빠른 경연시간을 그 마이크로 마우스의 공인 경연시간으로 측정한다.

④ 마이크로 마우스는 15분간 가능한 한 여러번 주행을 시도할 수 있다.

⑤ 마이크로 마우스가 도착지점에 도착한 후 제작자가 마이크로 마우스를 집어낼 수 있으며 이 경우는 그 이후의 모든 주행에 대하여 10초 삼각 규정을 적용받지 않는다.

⑥ 주행시간을 시작점과 종착점에서 수평으로 설치된 적외선 감지기를 이용하여 100분의 1초까지 측정되며 감지기는 바닥에서 1cm 높이에 설치된다.

⑦ 일단 주행 미로가 발표된 후에는 마이크로 마우스에는 미로에 관한 정보를 입력할 수 없다.

⑧ 조명, 온도, 습도 등은 정상적인 수준으로 유지되며, 경기중 변화시킬 수 없다.

⑨ 만일 마이크로 마우스가 오동작을 하는 경우에는 제작자는 경기를 중단시킬 수 있으며 그 경우 그 주행은 무효가 된다.

⑩ 만일 마이크로 마우스가 기술적인 문제로 조정이 필요한 경우 제작자는 허가를 받아 일정시간 이후 경기를 계속할 수 있으며 이 경우 벌점으로 3분을 소비한 것으로 간주한다.

⑪ 만일 마이크로 마우스의 일부를 교체하는 경우, 예를 들어 EPROM이나 건전지를 교환하는 경우에는 마이크로 마우스의 기억된 내용을 모두 소거해야 한다. 그러나 센서 조정과 같은 경미한 조정을 할 경우에는 기억 내용을 소거할 필요가 없다.

⑫ 마이크로 마우스의 어떠한 부분도 다른 마이크로 마우스에 이식시킬 수 없다.

⑬ 국내외에서 상품으로 개발 제작된 로보트는 대회에 참가할 수 없으며 심사위원장은 필요하다고 인정되면 마이크로 마우스의 세부를 조사하여 밝힐 권한이 있으며 주행을 중지시키거나 무효화 시킬 수 있다.

# 장애자올림픽 경기 운영시스템
# 프린스(PRINS)

취재/이항선 기자

전세계 장애자의 가슴에 참된 용기와 극복의 꽃을 피우고자 시작되어 도전과 극복, 평화와 우정, 참여와 평등을 그 이념으로 참가국 65개국, 참가인원 5,798명, 대회운영요원 12,216명, 행사 출현요원 9,592명, 대회종목 15개 종목(세부종목 729개)이란 사상 최대의 규모로 제8회 장애자 올림픽이 지난 10월 15일부터 24일까지 서울에서 개최되었다.

이번 대회는 장애자에 대한 국민의 인식을 높이고 세계 장애자간의 우의와 친선을 도모하는 대회, 기존시설 물자들을 최대한 활용하는 검소한 대회로 한국인의 정성을 심어주고 우리문화를 소개하는 특성있는 대회로 장애자올림픽 사상 유례없는 가장 성대하고 내실있는 대회였으며, 또하나의 개가는 역대의 어느 장애자올림픽에서도 사용되지 않았던 새로운 전산시스템이 도입되어 복잡하고 다양하며, 기록위주의 타 경기대회와는 다른 장애자올림픽에 대한 정보를 신속하고 정확하게 전달할 수 있었다는 것이다.

이 전산시스템중에 삼성데이타시스템에서 개발하고 지원한 경기운영시스템(프린스)는 경기결과 입력처리, 경기전 스타트리스트 출력, 경기결과 기록표를 산출, 차기대진입력, 모든 경기결과의 수집, 보관, 유지, 제공을 주요기능으로 하며 참가선수들의 장애유형 또는 등급에 따라 경기규정이 복잡하고 한 선수가 여러 종목에 참여할 수 있다는 장애자올림픽의 특수성에 맞추어 장애변경 프로그램 기능을 강화하였다.

## 4세대언어인 포커스 채택

프린스는 장애자올림픽과 같이 변경사항이 속출하는 정형화되지 않은 업무를 처리할 수 있게 자료처리시 제공되는 방법이 많고 그 논리가 간단하고 순간적인 처리에 유능한 4세대언어인 포커스를 채택하여 릴레이션 데이터베이스로 관리하고 온라인 리얼타임 방식을 이용하여 종합정보망에 연결하였다. 이렇게 하여 사상 최대의 규모를 메뉴선택방식과 도움말, 색인 등의 각종 기능을 활용하여 간편하게 조작할 수 있고 신속한 정보를 전달하며, 자료검증이 용이하도록 되어 있다. 또한 원격발생이 가능하며, 최적의 시스템이 유지되어 자료의 통계화와 함께 안전한 정보를 유지할 수 있게 하고, 역대의 자료를 통합하여 다양한 정보를 제공하고 경기결과가 자동으로 보고되도록 하는 특성을 가지고 있다.

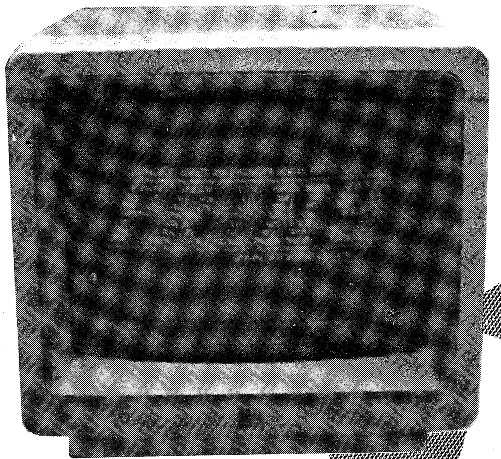
또한 짧은 개발기관과 적은 비용으로 전산화가 시도되어 대회규모에 적합하도록 중앙집중처리시스템을 구축하여 정보를 통제하거나 관리하는데 용이하게 되어 있다.

시스템 전체의 구성은 IBM4381 대형컴퓨터 2대와 여기에 워크스테이션 80대를 연결시켜 중앙에서 정보를 통제하도록 되어 있다. 그림 1은 전체적인 시스템구성을 나타내며 여기에서 사용된 장비는 표 1에서 보는 바와 같다.

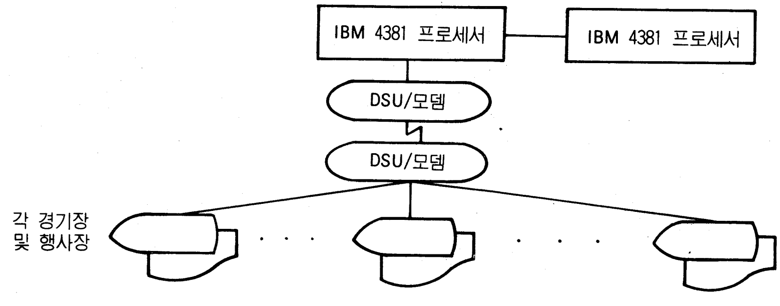
## 정보관리 4가지로 구분

프린스는 경기결과처리 및 경기운영지원, 정보제공, 선수등록관리로 구성되어 있다.

경기결과처리 및 경기운영지원은 종목별로 경기결과를 처리하고, 일정 및 대진표를 제공하고,



〔그림 1〕 시스템 구성도




〔표 1〕 사용장비

| 사 항 | 품 명 | 댓수 |
|---|---|---|
| 본체(CPU) | IBM 4381 | 2 |
| 기억장치(MEMORY) | 디스크<br>테이프<br>카트리지 | 8<br>2<br>2 |
| 프린터(PRINTER) | 레이저 프린터<br>라인 프린터<br>로컬 프린터 | 4<br>2<br>80 |
| 단말(TERMINAL) | 콘솔<br>터미널 | 4<br>80 |
| 통신장비 | 컨트롤러<br>DSU<br>MODEM<br>MSD<br>통신회선<br>통신카드 | 4<br>122<br>4<br>2<br>63<br>80 |

경기일정을 관리하고, 경기결과와 기록을 조회할 수 있게 하였고, 장애판정에 대한 변경을 처리하게 하였다.

정보제공은 장애의 종류, 경기장 내역, 경기종목 안내, 선수 및 임원의 신상 조회, 경기결과, 메달 조회, 신기록 조회, 세계기록 조회, 일정안내 등이다.

선수등록관리는 각종 코드등록, 선수 및 임원등록, 참가자 통계현황, 국가별 종목별 참가자, 장애별·등급별·직책별 참가자, 세계기록등록에 대한 사항을 관리하도록 되어 있다.

## (1) 등록처리

각국의 장애자 스포츠기구인 NSO에서 3차 엔트리가 접수되어 세계장애자기구인 IOSD에서 올림픽과 세계기록을 제공받아 서울 장애자올림픽대회조직위원회인 SPOC 운영과에서는 탈락자와 탈락종목을 다시 각국 NSO에 통보하고 3차 엔트리를 통해 선수화일을 만들어서 통계화일에 저장하고 자료 산출을 한후 SPOC에 전달한다. 한편 세부종목을 확정하는 작업에서 탈락자에 대한 정보를 SPOC 운영과와 선수 임원화일에 전달하고 확정된 작업을 종목별 참가자 화일에 저장한다. 그리고 이것을 다시 기록화일로 만들어 저장하고 종목별 참가자가 일정과 대진표화일로 입력되면 SPOC 담당관에게 세부종목별 참가자가 알려지고 SPOC 종목 담당은 일정 및 대진을 다시 입력한다. 이렇게 하여 일정 대진이 기록화일로 전달된다.

## (2) 장애판정

장애자올림픽은 일반올림픽과 달리 장애종류 및 등급에 따라 경기를 치룬다. 일반올림픽의 복싱이나 레슬링 등의 체급별 경기에서 헤비급 선수와 플라이급 선수가 똑같은 조건으로 경기를 치룰 수 없듯이 장애의 심도에 따라 등급을 두어야만 같은 조건에서 시합을 하게 되는 것이다. 그러므로 장애자올림픽에서는 엄격한 의무분류를 하여야 하므로 장애판정에 따른 변경처리가 중요하다.

장애등급 변경처리 과정시에 장애판정용 자료를 제공하여 세계 장애자스포츠기구 국제 조정위원회인 ICC에 소속된 장애판정관이 장애판정을 처리하면 변경된 자료를 선수 임원화일에 저장하고, 변경자료는 세부종목 변경에 대한 처리에 입력되고, 종목수정용 자료가 SPOC 운영과에 전달되어 종목변경자료가 다시 세부종목 변경처리 과정을 통해 선수 임원화일에 저장된 후 통계화일로 저장된다. 또한 세부종목을 확정하여 종목별 참가자 현황 화일로 저장하고, 참가자에 관한 현황을 SPOC 종목담당에게 전달하여 일정과 대진을 수정케 하여 일정과 대진을 기록화일에 저장한다.

## (3) 결과처리

경기장의 기록원이 결과기록지를 종목담당관에게 전달하면 결과기록은 경기결과 기록입력 처리를 통해 출전선수 명세와 결과기록지 확인자료가 다시 종목담당관에게 전달되어 변경처리할 수 있는 자료를 제공하여 일정과 대진을 작성하게 한다.

경기결과기록 입력과정을 통해 기록이 일정과 대진 처리과정에 전달되고 기록화일에는 경기결과기록과 변경처리된 결과를 저장하고 메달수를 처리한 메달현황과 결과기록지가 복사, 배포되어 담당관에게 전달되고 메달화일에는 메달현황이 처리된 내용이 저장된다.

이렇게 처리된 경기결과는 선수 및 심판등록 정보와 관련된 리스트, 대진

표 및 스타트리스트를 제공하고 경기결과(중간결과 포함)를 처리하여 공식적인 확인 과정을 거친다. 또한 단위결과의 종합(순위처리 등) 자동대진 또는 출전사항 처리에 의한 경기운영을 지원하며 전산센터의 호스트컴퓨터에 경기결과를 온라인 전송한다.

그 다음 결과를 소요 부수만큼 복사하여 결과 분배팀에 스타트리스트, 최종결과, 처리보고, 각종 통계 등을 전달하고 매일의 뉴스와 프로그램 발간부서에 소요된 결과를 제공한다.

### (4) 정보제공

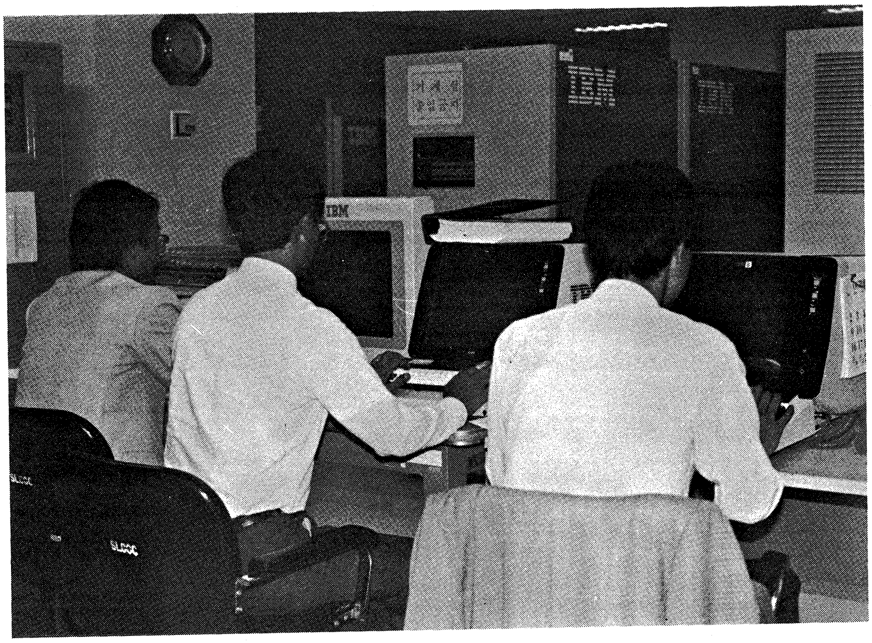
정보의 제공분야에서는 대회개요, 참가국 현황, 장애 종류, 종목 조회, 경기장 안내 등의 대회에 관한 정보와 일자별, 종목별 및 세부 종목별로 경기일정 및 대진표 상황과 선수신상, 임원신상, 개인기록에 관한 선수 및 임원신상에 관한 정보와 종목별과 국가별, 국가별과 장애별, 장애별과 종목별, 국가별과 직책별 참가자에 관한 정보와 국가별과 개인별, 국가별과 종목별, 세부종목별, 금메달 수상자, 다관왕에 대한 메달 현황과 경기결과 기록을 제공한다.

정보는 화면조회 처리과정을 통해 프레스 센터, 일반인, 임원과 각국 선수단이 관련된 리스트를 제공 받도록 하였다.

## 프린스의 운영절차

### (1) 경기전

- 선수 및 심판번호, 성명, 국가, 임무, 세부 출전사항 등의 선수단과 심판의 등록 명부에 대해 최종확인을 점검한다.
- 세부 경기일정 및 출전사항 확정에 따른 데이터처리로 경기일정표/대진표와 스타트리스트를 작성하여 경기부서에 제공한다.
- 최종적으로 경기결과의 정보전달 처리과정을 확인시키기 위해 교육 및 예행연습을 실시한다.




장애자올림픽의 모든 정보를 보다 빨리, 보다 정확하게, 보다 안전하게 처리 전달하기 위해 설치된 전산운영센터 상황실

### (2) 경기중 및 경기후

- 경기의 진행 직전 및 중간과정에서 발생되는 변경 사항을 처리한다.
- 경기의 진행에 따른 결과를 경기부서의 기록담당으로부터 전달받아 입력감독자가 코드 및 기재사항의 이상 여부를 1차 확인한다.
- 입력운영요원이 경기결과를 입력처리한 후 1차 출력에 대해 경기부서의 확인을 받는다. 확인이 끝나면 출력감독자에게 통보하여 해당 경기결과 출력을 프린트한 후 발생현황을 기록하고 복사팀에 전달한다.
- 복사팀은 각 출력별 소요 부수만큼 복사하여 복사조장이 분배를 위해 분류, 기재한 후 분배팀에 전달하여 각 부서에 공식 경기결과가 신속하게 제공되게 한다.
- 마감(수시 또는 매일)처리에 의해 관련되는 결과에 대한 요약, 일정/출전/대진 사항 변경, 종합결과, 각종 통계, 메달/신기록 발생사항 등의 하일라이트 출력을 프린트한다.
- 이의 출력을 복사후 분배팀에 전달한다.

### (3) 비상시 운영절차

비상사태는 하드웨어와 소프트웨어 장애, 변경처리 요구, 기타의 상황으로 발생한다는 전제아래 하드웨어는 정전과 단말기 고장, 통신기기 고장의 상황에서 소프트웨어는 시스템 소프트웨어인 포커스, 결과처리, 정보제공의 이상 발생으로, 변경처리 요구는 등록선수의 추가, 취소, 신상변경 및 선수 장애판정, 심판변경 등으로 인한 등록변경, 일정과 기록변경의 상황에서 비롯되며 기타는 자원봉사요원의 부족으로 인한 상황으로 보고 처리하도록 되어 있다.

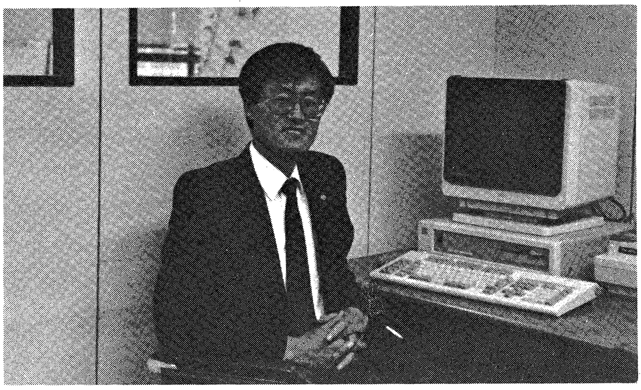
비상시 운영절차는 다음의 과정에 따라 처리된다.

- 장비를 점검하고 설치장소의 이상 유무를 확인한다.
- 장비통신 상태를 확인한다.
- 장애신고가 들어오면 전송회선 구간의 장애를 점검하여 이상이 있으면 백업라인을 교체한다. 이상이 없으면 시스템 이상을 확인한다.
- 시스템에 이상이 있으면 백업 포트를 교체하고 없으면 설치장소의 장애 발생을 살핀다.

• 설치장소에 이상이 있으면 전산 운영 센터인 OCC에 조치를 의뢰하고 없으면 장비를 확인하고 복구조치를 행한다.
• 이상으로 장애 복구가 완료되면 설치 장소에 통보하고 확인을 받는다.
• 여기까지의 운용과정을 작성한다.

인간의 평등을 확인하는 대회이며 인간 능력의 한계를 뛰어 넘는 감격의 장인 장애자 올림픽을 지원한 전산시스템이 우리 기술의 힘으로 개발되어 성공을 거둠으로써 과학기술한국의 이미지는 세계의 곳곳에 심어졌고 국내 전산화가 사회 전반에 걸쳐 이루어지고 있는 이때 장애자올림픽은 새로운 자신감을 얻을 수 있게 된 계기가 되었다고 할 수 있다. PC

**●인터뷰●/유광희/삼성데이타시스템 소프트웨어 개발부 연구원**



## 복지올림픽을 실현한다는 전제아래 세계적 차원에서 개발하여야

"무엇보다도 세계적인 프로젝트를 맡아서 해내야 한다는 어려움도 있었지만 개발요원들의 밤을 지새는 단합된 노력으로 이룰 수 있었고 특히 장애자올림픽 경기지원시스템으로 사회를 위하여 무엇인가 보탬이 될 수 있었다는데 보람을 느낍니다"라고 힘주어 말하는 유광희씨는 프린스 개발에 참여했고 현재 삼성데이타시스템 소프트웨어 개발부에 근무하고 있다.

장애자올림픽의 진행방식이 기록위주가 아닌 친교위주이기 때문에 장애판정이라든지 경기규칙이 경기 시작전까지 완료되지 못해 정확한 데이터를 적용할 수 없어서 곤란을 겪었지만 동료들의 격려와 개발요원들의 단합된 의지로 극복할 수 있었고 또한 선수촌을 방문했을 때 장애자들이지만 한결같이 밝고 행복해 보이는 선수들의 모습을 보고 자신을 되돌아 보고 반성할 수 있었던 좋은 경험을 가졌다고 한다.

"장애자올림픽 전산시스템이 더 발전하려면 우선 첫째로 경기 방식이 확정되어 업무가 정형화되어야 하고, 장애 판정 시기가 경기시작 2, 3일 전까지 확정되어야 합니다. 둘째는 기록경기와 대항경기 등 종목별 특성에 맞는 전산화가 이루어져야 합니다. 또 한가지 덧붙인다면 대형시스템을 사용하지 않고도 미니급 정도의 시스템에서 훌륭한 소프트웨어를 개발하여 비용을 절약하고, 장애올림픽의 전산시스템은 복지올림픽을 실현한다는 전제아래 세계적 차원에서 개발해야 한다고 생각합니다"라고 굳이 개인적인 의견이라고 고집하는 유광희씨의 말은 우리 모두의 바램이 아닐런지.

### 〈용어해설〉

**●패스워드(Password)**
패스워드는 프로그램, 오퍼레이터 또는 유저마다 따로 정해진 문자 스트링으로 구성되는 신호로서 기밀정보 또는 프라이버시에 관한 정보를 보호하는 역할을 한다.
데이터를 꺼내기 전에 이것을 사용하며 구체적으로는 시분할 시스템에서 단말장치를 사용할 권리, 화일내의 데이터를 이용할 권리를 나타내기 위해 각 개인에게 부여된다.

**●세션(Session)**
회화식 시분할 시스템에서 이용자가 회화를 위해 시스템에 접속하여 로그 온(log on)의 회화개시부터 로그 오프(log off)의 회화종료까지 지속된 회화처리 단위를 말한다.

**●서브루틴(Subroutine)**
프로그램의 일부분으로 문제의 일부를 계산하기 위해 그 자신이 독립된 루틴.
서브루틴은 호출 명령에 의해서 서브루틴이 실행된다. 서브루틴이 종료되면 다시 주 프로그램의 호출 명령 다음 문으로 돌아간다.

**●코루틴(Coroutine)**
메인루틴과 서브루틴 같이 주종 관계가 아닌 대등한 관계로서 상호 호출하게 되는 루틴이다. 예를 들면 게임 프로그램 등이다.

**●파라미터(Paramiter)**
프로그래밍할 때는 결정하지 않고, 실행할 때에 명령의 세부 동작을 구체적으로 지정하는 문자나 숫자를 말한다. 예를 들면 분류의 키를 지정하는 것 등이다.

**●세그먼트(Segment)**
프로그램이 너무 커서 주기억장치에 수용할 수 없는 경우, 적당히 분할하는 단위가 세그먼트이다.
세그먼트는 주기억장치와 보조기억장치 사이에서 부분 로드가 가능하게 하여 프로그램을 실행시킨다.

**●버퍼(Buffer)**
데이터를 어떤 장치에서 다른 장치로 전송할 때 양자 사이에 존재하는 데이터 전송속도의 차이, 또는 어떤 현상에서 나타나는 시간적 차이를 보완하기 위해 일시적으로 데이터를 보관해야 한다.
이와 같이 입·출력 조작을 위하여 일시적으로 준비된 기억영역이 버퍼이며 여기에 데이터가 임시로 보관된다.

**●덤프(Dump)**
기억장치 내용의 전부 또는 일부를 기록매체에 카피하는 것을 가리킨다. 보수 유지나 디버그(Debug)하기 위해 인쇄물로 하여 기억 장치의 내용을 출력하는 경우에도 사용한다. 출력형식은 2진, 8진, 16진의 ASCII코드 등을 사용한다.

# 워드 프로세서 세계의 신기원을 이룩한 젬워드

OA라는 용어가 국내에 소개된 이래 이제 사무자동화는 최첨단의 시대로 접어들고 있다. 작게는 전자타자기에서부터 복사기, 팩시밀리, 전자칠판 나아가 PC 및 워드프로세서 전용기에 이르기까지 그 수요가 날로 증가하고 있다.

특히 현대사회에서 쏟아져 나오는 각종 서류, 문서 등을 효과적으로 처리, 보관하기 위해 보다 기능적이고 효율적인 워드프로세서 전용기가 더욱 절실히 요구되고 있다. 워드프로세서는 기존의 문서작성기인 타자기보다 글자체가 미려하고 다양하며 능률적인 고도의 기능을 갖고 있어 전문 출판, 편집인 뿐만 아니라 일반 사무실에서도 크게 각광을 받고 있다.

근래에 와서 이 시장에 참여하는 국내업체의 수가 늘어가고 있으며, 또한 사용자측에서도 워드프로세서의 장점을 인식하여 어느 장소에서든 간편히 업무를 처리할 수 있는 이 기계에 대해 많은 관심을 보여 워드프로세서 전용기 시장이 활성화를 보이고 있다. 더우기 단순한 문장작성 기능만을 가지고 있던 초기의 워드프로세서 전용기에 만족하던 사용자들이 점차 나은 기능 즉 스프레드 쉬트 및 데이터 베이스 등의 기능 추가를 요구하게 되고 또한 출력된 글자의 미려함에 신경을 쓰게 됨으로써 이제는 고품질의 하드웨어와 함께 다양한 소프트웨어 제공이 앞으로의 워드프로세서 전용기의 시장에 커다란 판매요인으로 작용될 것이다.

책상위에는 컴퓨터가 한자리 차지하고 있다. 그 위에 모니터가 놓여 있고 또 그 옆에 적지않은 자리를 차지하고 있는 프린터, 그리고 서류들은 모니터 위에? 요즈음 사무실에서 흔히 볼 수 있는 광경이다. 그러나 젬워드는 자그마한 본체 한대면 충분하다. 별도의 주변기기가 전혀 필요없다.

국내에서 새롭게 소개된 문서작성 전용 OA기기인 젬워드는 기존의 워드프로세서와는 달리 DTP(Desk Top Publishing) 개념을 도입, 독자적인 편리성을 갖춘 개인용 워드프로세서 전용기이다. 문장, 그림, 표, 그래프 등을 손쉽게 작성해 원하는 위치에 자유자재로 배치할 수도 있으며 언제든지 간단히 인쇄해 낼 수 있어 편리하다. 또한 내장되어 있는 3.5인치 FDD를 이용해 자료의 보관, 호출 및 검색이 용이하며 이미 작성되어 있는 문서에 대한 재편집 기능이 탁월하다.

## 구성

젬워드의 커버를 벗겨보면 제일 먼저 윗부분을 덮고 있는 프린터 메카니즘이 보인다. 젬워드는 모델에 따라 두종류가 있는데 젬워드-G의 경우에는 메인보드에서 직접 제어되는 열전사방식의 24핀 서멀 프린터를 채용하여 일반용지의 인쇄는 물론 OHP용 필림의 인쇄가 가능하며 젬워드-GT에는 본체와의 비트화상 방식을 이용한 24핀 임팩트 도트 매트릭스 프린터가 부착되어 초당 80자까지의 고속인쇄와 자동용지설정 기능을 보유하고 있다. 계속해서 프린터를 제거하고 나면 젬워드를 구성하고 있는 각 부분들이 한눈에 들어 온다.

**사진 1**에서 보는 것처럼 9인치 CRT를 기준으로 우측에 CRT 콘트롤보드가 있다.

640*400의 고해상도로써 도형작성 및 이미지 스캐너를 통한 그래픽 처리가 우수하며 또한 Paper White 타입이기 때문에 일반 PC에서 사용하고 있는 모니터에 비해 눈의 피로를 막아주고 40자*20줄의 화면을 구성하여 지금까지의 액정모니터와는 달리 문서의 레이아웃을 한눈에 확인할 수 있다. CRT 좌측의 3.5인치 FDD는 메인보드의 FDC(Floppy Disk Controller) 로직과 함께 IBM 호환의 양면 배밀도 레코딩 포맷을 제공한다. 그리고 각 주요부분에는 전원을 공급해주는 전원부가 있다. 그러면 메인보드는? 이제는 젬워드의 아래쪽 커버를 벗겨내야 한다.

## 메인보드의 구성

젬워드의 주요부분인 메인보드는 고속처리 기능을 위한 16/32비트 마이크로프로세서인 CMOS 68000과 384K바이트의 램, 1.38M바이트의 롬 그리고 각 주변장치의 제어를 위한 게이트 어레이(Gate Array) 및 관련 칩들로 구성되어 있다. 다음 **그림 1**은 하드웨어의 구성을 그림으로 나타낸 것이다. 그러면 좀 더 구체적으로 각 부분별 특징 및 기능을 살펴보자.

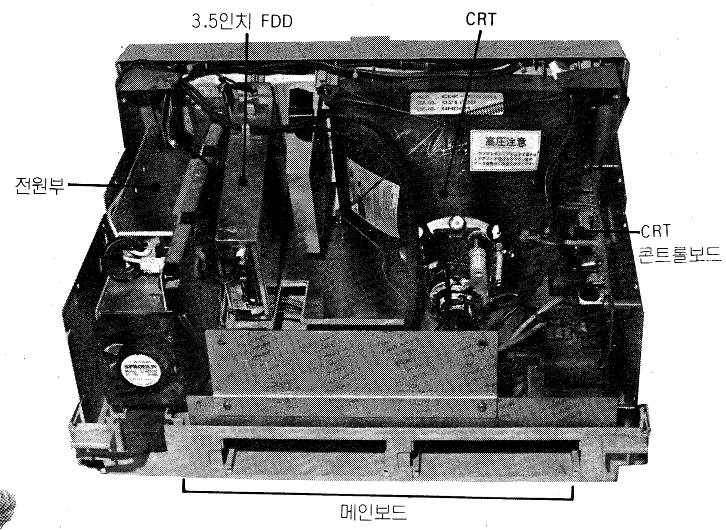
### (1) CPU

68000은 24비트의 어드레스 버스를 이용, 최대 16메가 바이트까지의 직접 어드레싱(Direct Addressing)이 가능



젬워드의 외관

〔사진 1〕 젬워드의 분해도




하기 때문에 2MIPS(Million Per Second)의 처리량 및 비동기 인터럽트 방식의 채택과 함께 방대한 프로그램을 빠른 시간내에 처리하여 원하는 출력을 제공해 준다.

또한 80계열의 입·출력 액세스(Isolated I/O) 방식과는 달리 메모리 Mapped I/O 방식을 이용하고 있기 때문에 회로설계가 용이한 장점이 있다.

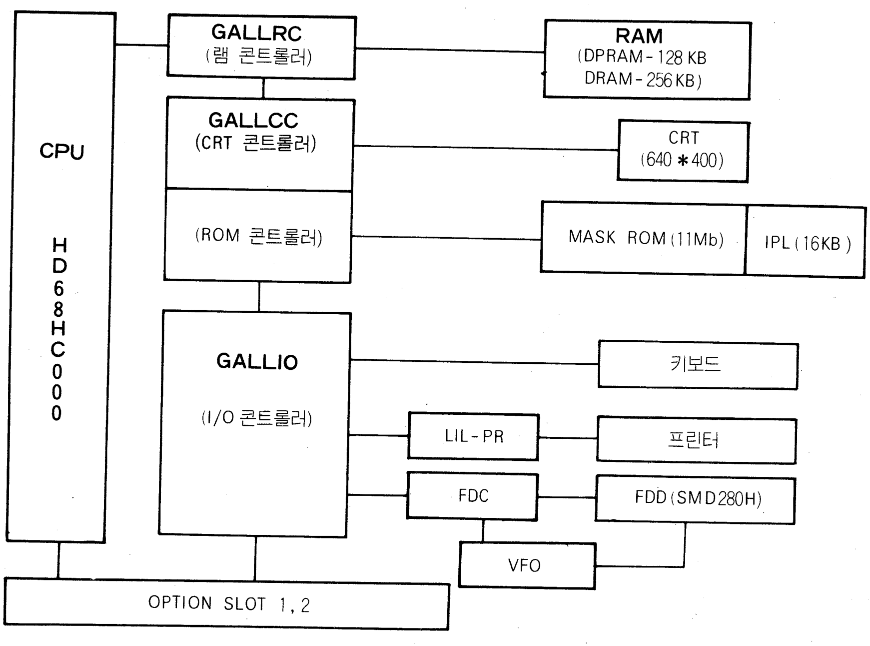
젬워드에서는 빠르고 다양한 텍스트 및 이미지 프로세싱을 위하여 모토롤라 68000의 High CMOS 타입인 HD68HC000(10MHz)을 CPU로 사용하고 있다. 시스템 클럭으로는 14.7456MHz 크리스탈을 카운터로 2분주하여 7.3728MHz을 사용하고 있으며 이것은 또 전송속도기준 클럭으로도 사용된다. 여기서 발생되는 대부분의 제어신호는 각 기능에 따라 해당 게이트 어레이를 통해 필요부분을 제어하게 된다.

## (2) GATE ARRAY

● **GALLCC** : CRT 제어를 위한 전용 칩으로서, HSYNC, VSYNC 및 비디오신호의 생성과 DRAM 메모리의 어드레스를 만들어 주면 ROM의 칩 선택을 위한 콘트롤러로도 사용된다.

즉, 데이터버스를 통해 비디오 램(DPRAM-Dual Port RAM) 영역의 데이터를 받아 내부의 시프트레지스터를 사용하여 CRT에 직렬 포맷으로 데이터(VIDEO신호)를 보내는데 이때 21.2 MHz의 도트 클럭을 기준으로 하는 카운터가 내장되어 있어 HSYNC 및 VSYNC신호를 제어한다.

〔그림 1〕 젬워드의 하드웨어 구성도



또한 DRAM에 대한 REFR(Refresh Request)를 생성하여 RAM전용 콘트롤러인 GALLRC에 전송한다.

**●GALLRC** : CPU(HD68HC000)와 DPRAM 및 DRAM과의 인터페이스에 이용되며 각 DRAM에 대한 제어신호를 생성한다. 특히 GALLCC로부터의 REFR(46.8μs에 1회 이상)을 받아 CPU 액세스와의 타이밍을 맞추면서 DRAM의 리플레쉬를 행한다.

**●GALLIO** : I/O에 관련된 각 부분의 제어에 이용된다. 즉 키보드제어, 프린트제어, FDD제어 및 각 주변장치로 부터의 인터럽트를 처리한다.

### (3) 메모리

**●RAM** : 젬워드는 총 384K바이트의 메모리를 가지고 있다. 비디오 램으로는 화면표시의 고속처리를 위해 32KB의 DPRAM(Dual Port RAM)을 4개 사용하고 있는데 그 중 64KB만이 VRAM 영역으로 할당되어 있다. 그리고 DPRAM의 나머지 부분은 32KB DRAM 8개와 함께 메인 메모리로 사용되며 따라서 메인 메모리의 사이즈는 총 320K바이트가 된다.

메인 메모리는 시스템 디스켓으로부터 상주 프로그램을 로딩시키기 위한 영역으로 약 96KB가 할당되어 있으며 또한 메모리의 최대 활용을 위한 오버레이 영역 및 기타 문서데이터 보관영역 등으로 이용된다.

한편 젬워드에서 최대 B4 사이즈까지의 문서를 작성할 수 있으나 이 문서가 모두 그래픽으로 채워질 경우 메모리 사이즈의 제한으로 인해 약 6부분으로 분리, 작성해야 하는 불편함이 있다.

**●ROM** : 시스템 프로그램을 RAM에 로드하기 위한 IPL ROM(Initial Program Loader ; 2764*2)과 화면표시 및 인쇄를 위한 C.G.(Character Generator)용 ROM 그리고 문서작성시의 한자변환에 이용되는 한자사전용 ROM이 있다.

C.G.용 ROM은 1M바이트 MASK ROM 3개와 4M바이트 MASK ROM 1개로 이루어져 있고 한자사전용으로는 4M바이트 MASK ROM 1개가 사용된다. 그리고 각 폰트의 구성은 다음과 같다.

3M바이트는 CRT 디스플레이용 10*10 폰트용이며 4M바이트는 프린터용 24*24 폰트 및 반자 처리를 위한 12*24 폰트 등이 내장되어 있다.

### (4) 키보드 제어

키보드 제어는 메인보드상의 게이트 어레이 GALLIO와 키보드 유니트내의 CPU-8748과의 인터페이스로 이루어진다. (그림 2 참조).

즉 키보드로부터 CPU(HD68HC000)로의 데이터 전송에는 8비트 전송버퍼가 이용되며 키보드는 이 버퍼의 내용을 전송라인 (KBDT, KBCK)를 이용해 직렬 전송을 한다. 키보드는 전송버퍼의 상태를 KBST 신호에 의해 인식한다.

키보드가 전송하는 데이터에 의해 버퍼가 차게되면 내부 플립-플롭에 의해 KBST 신호가 Low로 되고 이와 동시에 INTKB가 High로 되어 CPU에 인터럽트를 발생시킨다. 이 신호에 의해 CPU는 버퍼의 내용을 읽어들인 후, 인터럽트 신호를 리세트시키게 되고 이어서 KBST가 High로 되어 상기 동작이 반복된다.

### (5) CRT 제어

일반적인 CRT 제어로직인 CRTC회로가 별도로 있지 않고 전술한 바와 같이 GALLCC 내부에 포함되어 있어, 모든 CRT 콘트롤신호는 이 전용칩에 의해 생성, 제어된다. 비디오 램의 상태를 빠른 속도로 화면에 출력해 내기위해 입출력 단자가 분리되어 있는 DPRAM을 사용하고 있다. 비디오 램의 구성을 간단히 **그림 3**으로 나타내었다.

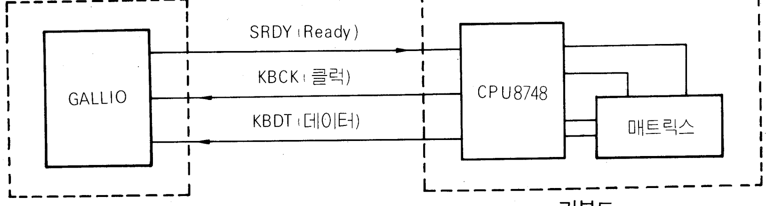
주 기능으로는 DISP ON/OFF기능, 수직 스크롤(화면 넘김)기능, 반전 표시기능 등이 있으며 가이드라인 표시 및 레이아웃 표시가 가능하다. 한편 CRT 도트피치는 그래픽모드에서는 1 : 1로, 문자표시시에는 1 : 1.2로 해서 사용된다.

### (6) 프린터 제어

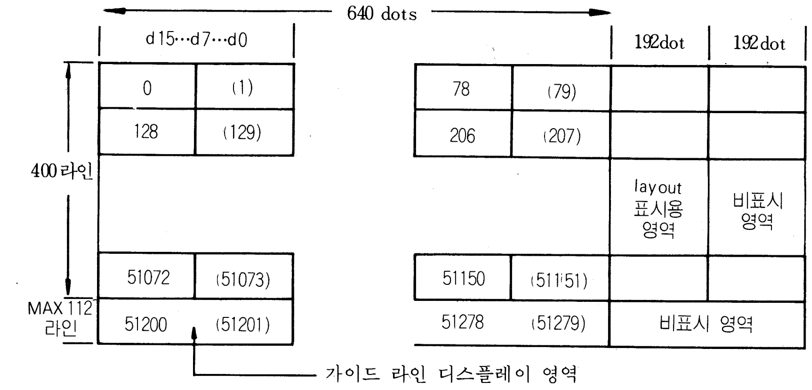
프린터는 젬워드-G와 젬워드-GT의 경우가 다르다. 우선 젬워드-G의 경우에는 프린터를 구동시켜 주는 펌웨어가 시스템디스켓의 프로그램내에 포함되어 있다. 따라서 프린터 콘트롤보드(LIL-PR보드)상에서 모터드라이버와 프린터 헤드 드라이버 및 관련 칩만이 있다.

그러나 젬워드-GT의 경우에는 프린터 메카니즘 및 콘트롤보드 자체가 독립된 하나의 프린터로서 작동할 수 있게 되어 있다. 즉 주변기기인 프린터로서 본체(메인보드)와 접속되어 있다고 표현하는 것이 옳겠다. 또한 GT의 프린터보드에는 별도의 CPU-7811과 프린터 콘트롤을 위한 프로그램 ROM이 내장되어 있다. 따라서 젬워드-G에 비해 프린터를 제어하는 펌웨어 프로그램 사이즈가 매우 적다. 한편 젬워드(G 및 GT)의 프린팅 방식은 모든 문서의 내용을 비트 화상방식으로 처리한다.

〔그림 2〕 키보드 제어



〔그림 3〕 VRAM의 구성




## (7) FDD 제어

FDD는 메인보드상의 FDC(Floppy Disk Controller)와 VFO 방식의 데이터 Separator에 의해 제어된다. 양면 배밀도로서 IBM과의 호환성을 가지고 있으며 최대 8인치 까지의 지원이 가능하고 또한 최대 4개까지의 플로피디스크 부착이 가능하나 현재 젬워드에서는 1FDD만을 제공하고 있다. 지원 가능한 모드는 다음과 같다.

| | | | | |
|---|---|---|---|---|
| MFM | 512 B/S | 9 S/T | 80 T/H | 2 H |
| MFM | 256 B/S | 16 S/T | 80 T/H | 1 H |
| FM | 256 B/S | 9 S/T | 80 T/H | 1 H |

B=바이트, S=섹터, T=트랙, H=헤드

## (8) 기타

젬워드에는 2개의 범용 옵션 슬롯이 있다. 현재 젬워드에서 제공되고 있는 옵션으로는 마우스, 이미지 리더, 통신세트 및 외부 프린터와의 인터페이스 등이 있다.

각 슬롯은 동일한 사양을 가지고 있기 때문에 어느 슬롯을 이용하여도 관계가 없으며 각 옵션에 대한 구분은 시스템 프로그램에서 처리하고 있다. 특히 통신 세트는 일반 PC와의 데이터 호환 및 요즈음 국내에서 보편화되고 있는 정보통신망(Network)의 사용을 위하여 별도 개발되었다.

젬워드는 RS-232C 방식을 채택하고 있는 타제품과의 인터페이스가 가능하다. 즉, RS-232C 방식을 이용한 통신세트가 개발되어져 있기 때문에 기본적으로 한글 및 한자 보드체계만 일치하면 데이터 호환이 가능하다. 젬워드는 기본적으로 KS-완성형 코드체계를 채택하고 있으나 아래와 같은 코드체계도 제공하도록 개발되어져 있다.

**(1) TriGem PC 2바이트 조합형**
**(2) KS-완성형**(각 업체마다 이 코드를 채택한 제품을 내놓고 있다.)
**(3) KS-7 멀티 바이트**(주로 XENIX 시스템—한글 시작/종료 코드는 사용자가 지정하게 되어 있음)
**(4) 데이터 서비스 정보망과의 접속**(H-MAIL 및 E-MAIL과의 접속 가능)

단, 상기의 데이터 호환은 PC의 그래픽처리와의 상이함으로 인해 텍스트 송·수신만이 가능하며 디스켓의 호환은 불가능하다. 그리고, 프린터를 사용함에 있어 유저가 더욱 빠른 속도를 원하거나 기존에 보유하고 있는 프린터를 이용하고자 할 때에는 제공되어 있는 옵션을 통해 인터페이스가 가능하다.

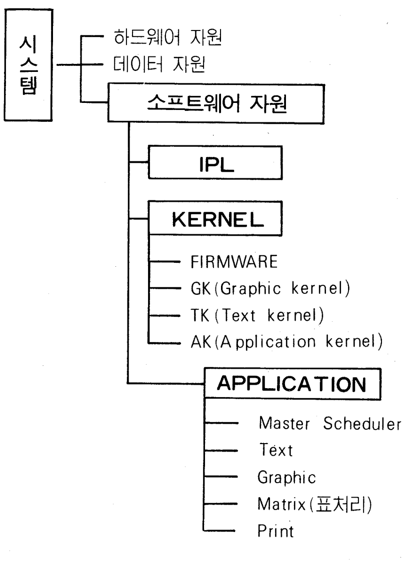
외부 프린터 사용을 위해서는 슬롯에 삽입할 인터페이스 보드와 케이블만 있으면 프린팅시에 메뉴 선택에 의해 손쉽게 이용할 수 있다(상기한 인터페이스 보드 및 케이블은 현재 옵션 판매중이다).

# 시스템 기동

다음에는 전원 투입후에 시스템이 기동되는 순서를 간단히 살펴보기로 하자. 우선 전원이 투입되면 시스템 전체가 리세트된 후 IPL 롬내의 프로그램에 콘트롤이 전달된다. 이어 IPL내의 프로그램에 의해 각 주변장치들에 대한 초기화 작업이 시작된다. 우선 메모리를 체크해 상태를 확인한 후 모든 메모리를 클리어 한다. 특히 비디오램 부분에 대하여는 특정값을 라이트해 VRAM 및 게이트 어레이의 상태를 체크한다. 이때 화면은 전체가 회색으로 변해 메모리 초기화가 정상적으로 이루어졌음을 나타내게 된다. 또한 옵션슬롯의 상태를 확인하여 해당 디바이스에 대한 처리를 행하는 등 각각의 초기화 단계를 진행한다.

이와 같이 초기화작업이 끝나면 FDD를 체크해 시스템 디스켓이 세팅되기를 기다린다. 시스템 디스켓의 세팅 및 확인이 끝나면 디스켓의 프로그램(Master Scheduler, 각 Kernel 등)을 읽어 램의 특정부분에 로드하게 되는데 이후로는 모든 콘트롤이 램에 로드된 Master Scheduler로 넘겨져 각 Firmware 및

〔그림 4〕 시스템 자원



Kernel의 초기화가 시작된다.

## 다양한 기능 발휘

**●한글, 한자, 영문, 일어 종합 워드 프로세서**

키보드상에 한글, 영문, 히라가나, 가다가나 키가 있어 임의의 키를 선택함에 따라 원하는 문자를 입력할 수 있다. 단, 한자의 경우에는 우선 한글로 입력한 후 '한자 변환'을 시켜야 한다. 한자변환에는 단어 단위로 변환시키는 '부분변환'과 일정량의 문장을 한꺼번에 변환시키는 '전체변환'의 두가지 방법이 있다.

'전체변환'시에 한글로 표시해야 하는(예 : 순한글, 조사) 부분까지도 한자로 변환이 되어 다소 불편하지만 약 5만 단어의 한자가 내장되어 있어 여러 동음의 한자중에서 유저가 원하는 뜻을 가진 한자를 적절히 유용하게 선택하여 사용할 수 있다. 그리고 986 가지의 특수 문자가 내장되어 있어 기술적인 전문용어를 사용하고자 하는 유저들은 유효 적절히 사용할 수 있다.

**●자유롭고 뛰어난 그래픽 처리**

젬워드에는 '구조'라 칭하는 3가지 종류의 모드가 있다. 즉, 문장작성을 위한 〈문장구조〉와 도형작성을 위한 〈도형구조〉 그리고 스프레드 쉬트 및 그래프 작성을 위한 〈표 구조〉이다. 각각의 구조는 나름대로의 기능을 최대한 집약시켜 놓았기 때문에 타기종에 비교하여 볼 때 뛰어난 기능을 가지고 있다.

특히, 도형구조에는 도형작성을 위한 각종 기능을 이용하여 자유자재로 그림, 도형 및 무늬작성을 할 수 있다.

하지만 크기가 일정하게 제한(최대 110×110㎜)이 되어 있어 다양한 크기의 '도형 구조'를 표현할 수 없으며 조작이 매우 불편하다.

**●사진까지 입력되는 이미지**

도형 작성시 키보드나 마우스를 이용하여 표현하기 힘든 그림이나 사진 등은 이미지 리더를 이용하여 간단히 입력할 수 있다. 물론, 입력된 내용을 확대, 축소 할 수 있으며 일부분만을 확대시켜 강조 할 수도 있다.

또한 문자수식의 확대나 축소시에도 1회 이상 사용할 수 있어 편리하다.

**●키보드가 필요없는 마우스**

젬워드는 전체가 메뉴방식으로 되어 있기 때문에 실제 문자를 입력하는 것만을 제외하고는 모두 마우스로 간단히 처리할 수가 있다. 특히, 도형구조에서의 마우스 사용은 도형작성에 소요되는 시간을 많이 절약할 수 있게 해 준다.

**●다양한 인쇄기능**

인쇄방식에는 〈표준인쇄〉와 〈고속인쇄〉, 〈일괄인쇄〉 그리고 〈화면인쇄〉 등이 있다. 표준인쇄는 출력문서의 글자체 질을 중요시 할 경우에 사용되며 고속인쇄는 빠른 속도로 출력하고자 할 경우, 그리고, 일괄인쇄는 두페이지 이상의 문서를 한번의 키로 출력하고자 할 때 사용한다. 물론 매수지정의 의해 여러매 인쇄도 가능하다.

또한, 프린터로 작성된 문서를 미리 인쇄상태로 측정할 수 있는 인쇄용 디스플레이 기능을 갖추고 있어 불필요한 인쇄는 사전에 방지하고 레이아웃을 한눈에 파악할 수 있다.

**＊정전시의 백업(Back Up)기능**

즉, 일정량의 데이터가 입력되면 디스켓에 자동 저장을 시켜 정전시에 대비하고 있다. 저장되는 양은 약 2K바이트 상당이다. 그러나 2K바이트 저장후, 다시 저장되기 전에 정전되는 상황에 대한 대비는 되어 있지 않다.

이상과 같이 젬워드는 다양하고 놀라운 기능을 가지고 있지만 멀티플랜 기능이나 그래픽 처리에서의 크기의 한정성, 화일관리가 불편한점 등 몇가지 기능의 보완이 필요한것 같다. 완벽한 기능을 발휘하는 젬워드를 기대해 본다.

### 제품사양

〔표시부〕

표시 방식:9인치 CRT(Paper White)

표시문자수:40열＊20행(640＊400 도트)

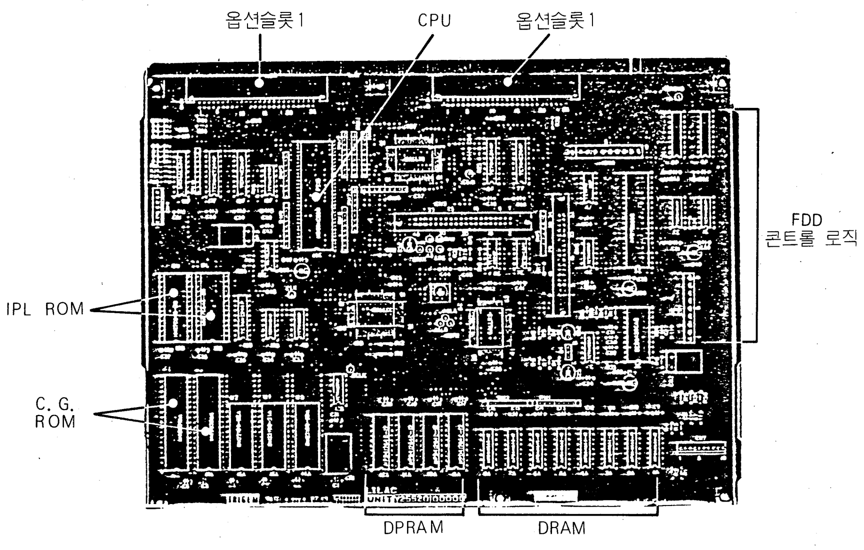
문자 구성 : 16＊16 도트

〔인자부〕

인자 방식:열전사 혹은 임팩트 도트 매트릭스 방식

인자 속도:젬워드-G;최대 37.5CPS

〔그림 5〕 메인 보드



젬워드-GT ; 최대 80CPS
문자 구성:24*24, 24*18, 16*16
인자체:명조체, 영문(Courier, Script)
용지 사이즈:엽서사이즈 부터 최대 B4 횡폭까지
〔기억부〕
내부 기억:384KB(A4 용지 50페이지 상당)
외부 기억:3.5인치(양면배밀도720KB) FDD(A4 용지 350페이지 상당)
〔입력부〕
방식 : 한글, 영수자 입력(한자는 변환 방식)
문자종류:KS-완성형
(한글 2350자, 한자 4888자, 특수문자 986자)
〔외관〕
치수 : 400mm(가로)*360mm(세로)* 220mm(높이), 약 10.5kg
〔전원〕
정격입력전압:AC 100V(입력전압범위: AC 90V—AC 110V)
주파수 : 49Hz—61Hz
소비전력:대기시—30W, 동작시—40W

# 노턴유틸리티 V.4.0

유현무/성대 정보공학과

컴퓨터 사용자가 가장 많은 시간을 소비하는 작업은 역시 프로그램 작성이다. 그러나 프로그램을 저장한 디스크를 관리하는 작업도 생각처럼 간단한 작업이 아니라는 것을 많은 사람들은 체험으로 익히 알고 있을 것이다. 이런 이유로 어떻게 하면 안전하고, 간편하게 디스크를 관리할 수 있는가 하는 문제는 대다수의 컴퓨터 사용자들이 가장 궁금하게 생각하는 점이다.

예를 들어 어느날 A군은 디스크의 화일을 정리하다 실수로 자신이 며칠 걸려 만든 프로그램 화일을 지워 버렸다고 하자. A군은 자신의 실수를 알았지만 이미 때는 늦었다. 모든 노력이 단 한번의 실수로 사라져 버린 것이다.

이와 같은 경험은 누구나 컴퓨터를 사용하다보면 한번쯤은 겪는 일이다. 어떻게 할 것인가. 모든 것을 포기하고 다시 시작할 것인가? 이런 경우 노턴 유틸리티를 사용하면 간단히 해결할 수 있다. 노턴 유틸리티는 디스크의 지워져 버린 화일을 복구할 뿐 아니라 엄청난 자료를 저장하는 하드디스크를 포맷시켜 버렸어도 다시 원상태로 복구시킬 수 있다.

노턴 유틸리티는 미국의 피터 노턴에 의하여 제작된 일종의 유틸리티 모음집으로 이 유틸리티는 PCTOOL과는 다르게 모든 명령을 하나의 화일로 처리한다(PCTOOL은 모든 작업을 PCTOOLS라는 프로그램 내부에서 실행한다). 따라서 어느 한 작업을 실행시키기 위하여 큰 용량의 프로그램을 전부 로드할 필요가 있다.

또한 기존의 유틸리티에서 볼 수 없었던 디렉토리 소트기능, 포맷된 하드디스크의 복구기능, 화일의 빠른 로드를 위한 디스크 정리기능, 그리고 편리한 디렉토리간의 이동기능을 갖춘 편리한 유틸리티이다. 여기서는 노턴 유틸리티의 여러가지 기능중 이러한 특별한 기능에 대해서만 설명할 것이다.

## 필요한 화일과 그 기능의 요약

노턴 유틸리티에서 필요한 화일과 기능을 요약해 보면 아래와 같다.

*DS.EXE…화일의 순서를 변경한다
*DT.EXE…디스크의 상태를 검사한다
*FA.EXE…화일 속성을 변경한다
*FF.EXE…디스크의 화일목록을 출력한다
*FR.EXE…포맷된 하드디스크를 복구한다.
*NCD.EXE…현재 디렉토리를 손쉽게 변경한다
*QU.EXE…지워진 화일을 복구한다
*SD.EXE…빠르게 프로그램을 로드
*UD.EXE…지워진 디렉토리를 복구
*VL.EXE…볼륨 레벨을 변경한다
DISKCOPY.COM…디스크의 백업을 위해
BACKUP.COM…하드디스크의 백업에 사용한다.

*표시는 NORTON UTILITY V.4.0 디스크에만 있다.

이외에도 TREEINFO.NCD와 RECOVER.DAT 등이 더 필요한데 TREEINFO.NCD는 NCD 실행시, RECOVER.DAT는 DAT는 FR을 실행시 이들 실행 화일에 의해서 자동으로 생성된다.

### (1) 디렉토리의 생성, 변경, 삭제

필자의 하드 디스크에는 디렉토리가 30여개나 있다. 그러나 필자 자신도 그 하드디스크 디렉토리의 트리구조가 어떻게 구성됐는지 잘 모른다. 어떤 디렉토리는 5단계의 CD(Change Directory) 명령으로 겨우 들어갈 수 있다.

```
/—LANGUAGE┬PASCAL┬Turbo—
          └C     └MS-PAS
— Source — N12
```

위와 같은 디렉토리중 N12로 들어가려면 DOS상에서는

```
C>CD\LANGUAGE\PASCAL\TURBO\SOURCE\N12 ↵
```

과 같이 긴 명령을 입력해야 하는 번거로움을 겪는다.

또한 N12에서 MS-PAS라는 디렉토리로 가고자 하면

```
C>CD\ ↵
C>CD\LANGUAGE\PASCAL\MS-PAS ↵
```

라고 입력을 해야만 한다. 이런 작업이 몹시 귀찮은 일이라는 것은 당연하다. 그러면 이런 작업을 손쉽게 해주는 NCD.EXE를 사용해 보자. NCD는 현재의 디렉토리를 손쉽게 변경시킬 뿐 아니라 디렉토리를 생성(Make Directory)하고, 삭제(Remove Directory) 시키는 것까지 가능하게 한다.

**NCD option directory-name /R**

- Option : NCD의 작동을 결정하며 MD와 RD가 있다.
  MD－디렉토리 생성시
  RD－디렉토리 삭제시
- Directory-name : 목적이 되는 디렉토리 이름과 경로(PATH)
- /R : TREEINFO. NCD를 재생성

**●디렉토리의 변경**

NCD 명령은 디렉토리를 변경시킬때 사용하며 두가지 방법이 있다.

첫째, 아래와 같이 드라이브명을 입력하고 **그림** 1의 화면상에서 막대를 원하는 위치로 움직여 디렉토리를 변경하는 방법이 있다.

**A>NCD C: ⏎**

둘째, NCD의 화면 그림 1을 이용하지 않고 직접 디렉토리 이름을 입력하여 디렉토리를 변경하는데 이때는 MS-DOS의 CD명령처럼 모든 경로를 입력할 필요없이 찾고자 하는 디렉토리의 이름 전부나 일부만 입력한다. 단, 이 경우 같은 이름 예를 들면 \NORTON\SD와 \UNIX\SD가 같은 디스크에 있을 경우 NCD SD라 입력하면 NCD는 에러를 발생하고 작업을 멈춘다. 따라서 NCD 사용시에는 같은 디렉토리 이름의 중복을 피하도록 해야 한다.

**●디렉토리의 생성**

NCD의 디렉토리 변경기능처럼 디렉토리의 생성시에도 두가지 방법이 있다. DOS 프롬프트상에서 MD옵션을 사용하거나 그림 1의 화면에서 M키를 사용한다.

그러나 첫번째 즉 MD옵션 사용시에는 DOS의 MD 명령과 다를바 없으나 그림 1 화면 사용시에는 매우 간편하게 디렉토리를 만들 수 있다. 만약

**A>MD\LANGUAGE\FORTRAN ⏎**

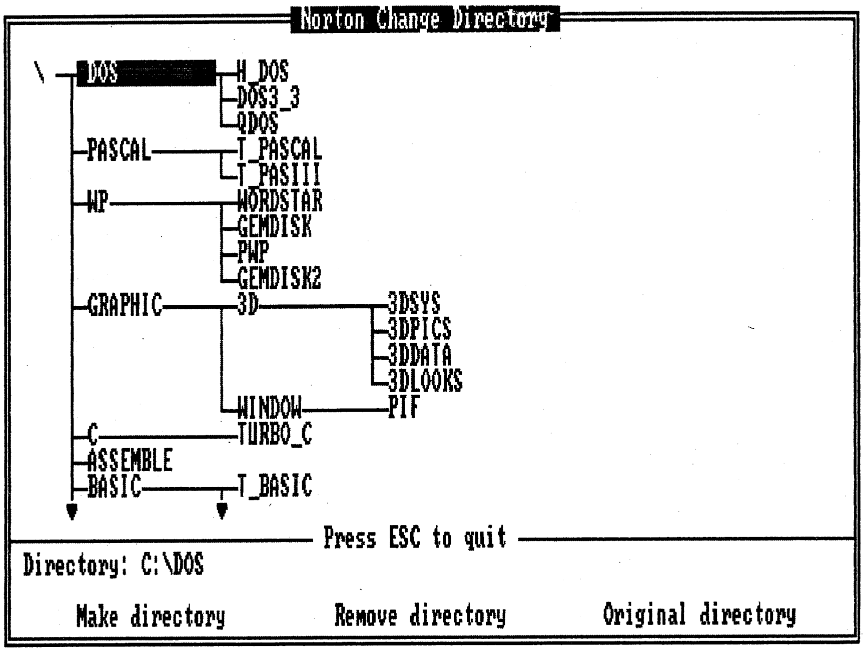
을 사용하여 LANGUAGE 디렉토리에 FORTRAN 서브 디렉토리를 만들었다면 NCD에서는 그림 1의 막대를 LANGUAGE에 고정하고 M키를 누르고 새로운 디렉토리를 입력하면 같은 효과를 거둔다.

**●디렉토리의 삭제**

삭제시에도 위와 같이 두가지 방법이 있다. 사용방법은 MD 명령과 같다.

NCD를 사용 않고 DOS상에서 MD, RD 명령으로 디렉토리의 구조를 변경시킬 경우 /R Parameter로 디렉토리 구조의 정보를 담은 TREEINFO. NCD를 변경시켜 주지 않으면 바꾸어진 디렉토리를 NCD가 알 수 없다.

〔그림 1〕 NCD의 실행




## (2) FORMAT된 하드디스크의 복구

A군이 새로운 디스켓이 필요하여 빈 디스켓을 포맷하기 위해

**A>FORMAT C:/S/V ⏎**

을 입력해서 포맷작업이 시작되었다. 그러나 잠시후면 자신이 어떤 엄청난 일을 벌였는지 알게 된다. 모든 화일 프로그램을 담은 하드 디스크가 모두 지워져 버린 것이다. 만약 이런 일이 어느 기업의 재무관리 자료용 하드 디스크였다면 엄청난 사고일 것이다. 그러나 걱정말라. FR. EXE로 그 문제를 해결할 수 있다.

대부분 DOS의 포맷 명령이 하드 디스크를 포맷할 경우 실제로는 데이터 부분은 파괴하지 않고 화일관리 영역만을 지워버린다. 따라서 하드 디스크의 복구가 가능하다.

**FR drive-name [/SAVE]**

- Drive-name : 복구시킬 하드디스크의 이름
- /SAVE : 하드디스크의 정보를 저장하는 RECOVER. DAT 생성

FR은 RECOVER. DAT라는 화일을 갖고 있어야만 포맷된 하드 디스크를 살릴 수 있다. 따라서 자주

**A>FR C:/SAVE ⏎**

으로써 하드 디스크에 대한 정보를 다른 디스크에 저장시켜야 한다.

FR은 하드 디스크를 대상으로 할 뿐이지 플로피 디스크는 복구할 수 없다는 것을 명심해야 한다.

## (3) 화일의 속성변경

SECRET. DAT라는 화일은 회사의 기밀을 담은 화일이다. 이 화일은 일반인은 보아서는 안된다. 따라서 이런 경우 어떤 방지책을 마련해야 하는데 어

**노턴 유틸리티의 FR.EXE 화일을 이용하면 포맷된 하드디스크의 내용도 안전하게 복구할 수 있다.**

떻게 해야 할까. 노턴 유틸리티는 매우 쉽게 이런 작업을 할 수 있다. 간단히 속성을 변경해서 보안을 유지할 수 있는 것이다. 즉 이런 때는 Read-Only 속성과 Hidden 속성을 이 화일에 첨가하면 된다.

화일의 속성에는 Archive, Read-Only, Hidden, System이 있는데 Archive 속성은 화일이 처음 생성시 갖는 속성으로 속성이 없는 것과 같다. Read-Only를 갖는 화일은 지워지지 않으며 화일의 변경이 불가능하다. Hidden 속성은 디렉토리상에서 보이지 않으며 다른 디스크로 카피할 수 없으나 로드하여 실행할 수는 있다. System 속성은 Hidden 속성과 같으나 로드가 불가능하다. FA는 이런 속성을 변경시키는 명령이다.

**FA file-name Attribute Parameter**

- File-name : 화일의 이름과 경로이다. 화일이름은 와일드 문자 *, ?를 사용할 수 있다.
- Attribute
  ① /A+ Archive 속성
  ② /HID+, /HID− Hidden 속성
  ③ /R+, /R− Ready-Only 속성
  ④ /SYS+, /SYS− System 속성
  화일 속성뒤에 +이면 속성이 ON, −이면 속성이 OFF된다.
- Parameter
  ① /CLEAR : 모든 속성이 사라지고 Archive 상태를 갖는다.
  ② /P : 출력화면을 페이지 단위로 한다.
  ③ /S : 서브디렉토리의 화일도 대상이 된다.
  /CLEAR를 첨가할 때는 다른 속성은 모두 무시된다.
  /A 속성은 다른 속성과 함께 사용할 수 없다.
  /HID+, /SYS+ 속성을 가지면 디렉토리상에 보이지 않으므로 주의한다.

IBMBIO.COM이나 IBMDOS.COM은 속성이 Read-Only, HID.SYS를 갖는다. 따라서 /CLEAR로 모든 속성을 제거하고 DEL 명령으로 지워버리면 그 디스켓으로는 부팅되지 않는다. 따라서 가능한 이 화일의 속성은 유지하는 것이 좋고 만약 지워졌다면 DOS의 SYS.COM으로 부팅 가능한 디스켓으로 만든다.

## (4) 디스크내의 화일 위치탐색

며칠전 하드 디스크의 어떤 디렉토리 안에 화일을 하나 만들어 두었는데 그 디렉토리가 생각나지 않았다. 그 많은 디렉토리를 모두 찾아봐야 할 것는가?

이처럼 하드 디스크 또는 디스켓 속의 잊어버린 화일을 찾는 것은 쉬운 일이 아니다. 이럴 때는 FF.EXE를 사용하여 쉽게 찾아낼 수 있다. FF는 DOS의 DIR 명령과 유사하나 DIR이 현재의 디렉토리 속의 화일만을 출력하는 반면 FF는 원한다면 하드 디스크 전체의 모든 디렉토리, 모든 화일을 출력한다.

**FF file-name Parameter**

- File-name : 찾고자하는 화일이나 디렉토리 이름, 경로를 입력한다. 이것이 생략되면 모든 디렉토리를 출력한다.
- Parameter
  /P : 화면 출력을 한 페이지 단위로 한다.
  /W : 한행에 5개의 화일을 출력한다.
  /A : 모든 드라이브를 찾는다.

화일명 입력시 와일드 문자 *와 ?를 사용할 수 있다.

FF를 사용하면 대상 디스크의 모든 화일목록을 임의의 화일로 만들어 저장할 수 있는데 그 방법은 **그림 2**와 같다. 이 방법으로 하드 디스크 모든 화일에 대한 목록을 정기적으로 만들어 쓰면 화일을 찾는다거나 중복된 화일의 삭제, 비어 있는 디렉토리의 제거 등 디스크 관리에 매우 도움을 줄 것이다.

## (5) 지워진 화일의 복구

화일이 지워진다는 것은 DEL 명령에 의하여 디렉토리 섹터중 해당되는 화일의 첫자를 삭제 코드로 변경시킬 뿐이다. 따라서 화일을 복구시킨다는 것은 변경된 첫글자를 다시 변경시키면 될 뿐이다.

**QU file-name /A**

- Filename은 경로와 화일의 이름을 입력한다. 이때 와일드 문자 *, ?를 사용할 수도 있다.
- /A 화일 복구시 첫자를 요구하는 메시지 없이 자동적으로 A, B, C… 순으로 채워나간다.

지워진 화일이 차지하고 있던 기억영역은 화일이 복구될 때까지 보존되지 않고 후에 다른 화일이 생성되거나 카피되어 기록되면 지워진 화일이 차지하던 영역을 침범한다. 따라서 다른 화일이 이 영역을 차지했다면 화일을 복구해도 실행시킬 수 없고 그 내용도 삭제되기 이전과 같지 않다.

/A를 사용하여 화일의 첫자를 A, B, C… 순으로 복구시킬 경우 만약 복구시킨 화일이 삭제 이전의 이름으로 다른 화일에 의하여 불리어져 사용된다면 화일이름을 이전의 이름으로 바꾸어야 하므로 이런 경우는 /A를 사용하지 않도록 하자.

〔그림 2〕 FF를 사용한 화일목록 저장

```
A>COPY CON TREE.BAT ↵
FF %1 >TREE.TXT     ↵
^Z
```

(a)

```
A>TREE C:           ↵
A>TYPE TREE.TXT     ↵
```

(b)

## (6) 프로그램의 빠른 리드

디스크를 여러번 사용하다 보면 지워진 화일과 새로운 화일의 크기가 맞지 않기 때문에 발생하는 미사용 영역 때문에 디스크내의 저장 상태가 매우 복잡하게 된다. 이 경우 프로그램의 리드나 로드시 시간이 많이 걸리는데 SD.EXE를 사용하여 미사용의 공간을 디스크의 뒷부분 섹터로 옮겨서 빠른 로드, 리드 속도를 얻을 수 있다.

```
SD Drive or file-name Parameter
```

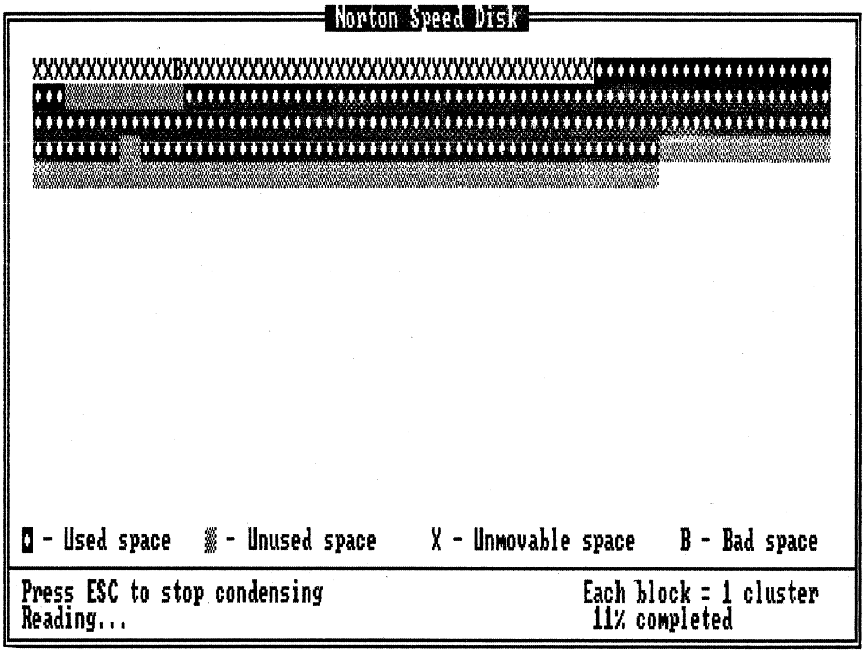
- Drive Filename : 대상이 되는 디스크나 화일을 지정.
- Parameter
  /Report : 화일 배치 상태만 출력
  /P : 페이지 단위로 출력
  /S : 서브 디렉토리의 분산 상태도 출력

SD를 실행하면 대상이 되는 디스크의 기억영역의 할당을 도식화한 맵화면 **그림 3**(a)이 출력된다. SD 실행 후에는 그림 3(b)와 같이 모든 미사용 공간이 디스크의 후방 섹터로 옮겨가게 된다.

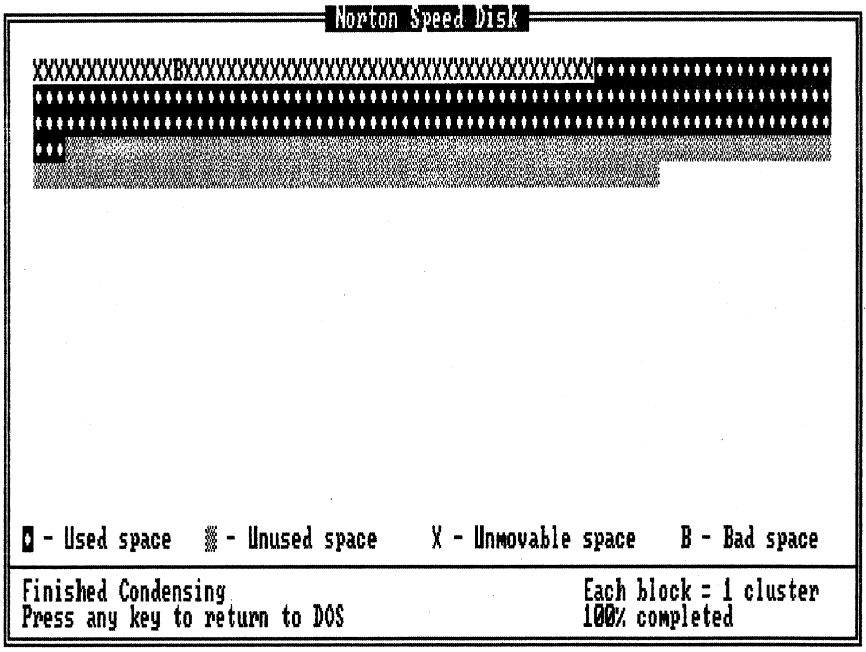
SD는 디스크의 기억공간의 배열상태를 바꾸기 때문에 SD의 수행도중 〈CTRL〉-C, 〈CTRL〉-〈BRK〉, 〈ESC〉를 누르면 작업이 정지되지만 디스크의 모든 데이터를 잃게 됨으로 주의한다.

일부 게임이나 프로그램들은 디스크의 특정한 섹터 위치를 바꾸어 복사방지책으로 사용한다. 이러한 특정 섹터의 위치가 변하며 그 프로그램을 복사할 수 없게 만든 프로그램에서는 SD를 실행하여 그 섹터를 옮기면 그 프로그램은 영구히 회복시킬 수 없다. 따라서 프로텍트가 걸린 디스크에 대한 SD.EXE의 실행은 삼가한다.

〔그림 3〕 SD를 실행하기 전




(a) SD를 실행하기 전




(b) SD를 실행한 후

## (7) 화일 순서의 변경

디스크 특히 하드 디스크의 경우 사용 횟수가 많을수록 각 디렉토리의 화일 배치가 엉망이 되어 버린다. 왜냐하면 어떤 화일을 지우고 새 화일을 추가하면 그 지워진 화일이 있던 공간을 차지하기 때문에 순서가 뒤바뀌기 때문이다. 이때는 DS.EXE를 사용하여 화일의 순서를 정렬해 보자.

```
DS Sort-key directory-name /S
```

- Sort-key : 화일 배치의 순서 원칙
  ① N : 화일이름(Name)
  ② E : 확장자(Extension)
  ③ T : 시간(Time)
  ④ D : 날짜(Data)
  ⑤ S : 크기(Size)
- Directory-name : 정렬하고자 하는 디렉토리와 그 경로
- /S : 대상 디렉토리의 서브 디렉토리의 화일까지 정렬

일반적으로 화일 배치의 순서는 화일

이름의 알파벳 순서와 확장자 순서로 하는 것으로 되어 있으나 필자의 경험에 비추어 볼때 아래처럼 대상 드라이브만 입력한뒤

예〉 A> DS B:

와 같이 DS의 메인 화면을 사용하여 아래 순서대로 정렬하는 것이 좋다.

① [COMMAND.COM]
② 배치 화일
③ 유틸리티 화일
(예 : HGC.EXE, NKP.EXE)
④ 주프로그램
⑤ 주프로그램의 부속화일
⑥ 사용자의 화일

이렇게 정렬할 경우 사용순서를 위에서 아래 방향으로 하기 때문에 프로그램 동작시 편리함을 얻을 수 있다.

DS는 속성이 Hidden이거나 System인 화일은 정렬할 수 없다. 따라서 이런 속성을 가진 화일은

```
A>FA 대상화일 /CLEAR ⏎
```

으로 속성을 해제하고 DS작업을 수행해야 한다.

## (8) 볼륨레이블의 새로운 변경

디스켓을 구별하기 위해 표면에 종이 레이블을 붙이는 것처럼 디스크의 내부에도 레이블을 기록하는 부분 즉 볼륨 레이블을 갖고 있다.

이 볼륨 레이블을 변경, 또는 작성하는 방법은 MS-DOS의 레이블 명령을 사용한다.

그러나 이 레이블 명령은 아래와 같이 많은 사용의 제한을 두고 있으며, 이러한 제한은 사용시 주의를 요한다.

- 영문자의 소문자는 대문자로 표시된다.
- #, - 등 몇개의 특수문자만을 허용한다.
- 공백을 넣을 수 없다.

이러한 제약을 받는 레이블 명령에 비하여 VL.EXE는 위의 모든 제약을 무시하는 볼륨 레이블을 만들어 준다.

```
VL drive-name "Volume Label"
```

- Drive-name : 대상 드라이브 이름
- "Volume Label" : 새로운 볼륨 레이블로 중간에 공백이 있을 경우 꼭 " "로 둘러싸야 한다.

만약 VL 드라이브 이름만 입력한다면 VL은 볼륨 레이블을 요구하는데 이때 " "는 생략해야 한다.

레이블 명령을 컴퓨터 바이러스((c) Brain)에 감염된 디스크에서 실행시켜도 볼륨 레이블은 바꿀 수 없다. 이것은 이 디스크는 바이러스가 양성 즉 다른 디스크로 전파가 되는 상태이므로 레이블 명령으로 바이러스의 양성 여부를 알 수 있다.

## (9) 지워진 디렉토리의 복구

화일이 지워진다는 것과 디렉토리가 지워진다는 의미는 비슷하다. 즉 화일이 Del 또는 Erase 명령으로 지워질때처럼 RD에 의하여 디렉토리가 삭제될 때도 디렉토리의 첫자만을 변경하게 된다. UD.EXE는 이 첫자가 변경되어 삭제된 디렉토리의 복구를 위해 사용한다.

```
UD path
```

- Path : 복구할 디렉토리의 상위 디렉토리 까지의 경로

예) 만약 C:의 \LANGUAGE\PASCAL\TURBO가 지워졌다면

```
UD\LANGUAGE\PASCAL ⏎
```

만을 입력한다. 그러면 지워진 첫자를 요구하는 메시지가 출력된다.

첫자를 입력하여 디렉토리가 복구될 때 만약 그 디렉토리가 화일을 갖고 있다면(물론 지워진 화일이다) 그 화일에 대한 정보가 출력된다.

지워졌던 디렉토리를 복구했다 하더라도 그 내부의 지워졌던 화일은 함께 복구되지 않는다. 이때는 QU.EXE를 사용하여 지워진 화일을 복구한다.

## (10) 디스크의 상태검사

사용하던 디스크에서 Reading Error, Writing Error 등이 나타날때 혹은 포맷후에 배드 섹터(Bad Sector)가 나타날 때 이러한 것은 디스크의 이상으로 사용자의 오류, 하드웨어의 문제에 의하여 디스크가 손상을 입었기 때문이다. 이 디스크는 그 손상된 위치에 따라 대응해 주어야 하는데 이런 작업을 해주는 것이 DT.EXE다. DT.EXE는 손상된 크기만을 알려주는 CHKDSK.COM과는 다르게 그 손상된 위치를 알려준다.

```
DT drive-name file-name Parameter
```

- Drive-name : 대상 드라이브의 지정
- File-name : 대상 화일의 지정
- Parameter
  /B : 디스크와 화일 검사를 동시에 수행
  /Cn : n번째 클러스터를 배드 클러스터로 바꾼다.
  /Cn- : n번째 클러스터를 정상상태로 바꾼다.
  /D : 디스크 검사만을 수행한다.
  /F : 모든 화일을 검사한다.

파라미터가 /B, /F일 경우는 화일 이름을 표시하는데 와일드 문자 *, ?

〔표 1〕 손상된 섹터와 그 증상

| 손 상 | 섹 터 번 호 | 증 상 |
|---|---|---|
| #0 | Boot Sector | Boot시 "Non system disk error" 발생. |
| #1~4 | File Allocation Table | 모든 작업시 "Reading Error" 발생. |
| #5~11 | Directory | 화일의 존재가 무시된다. |
| #11~40 | User Range | 화일의 내용이 손상된다. |

를 사용할 수 있다. 표 1은 손상된 섹터와 그때의 증상에 대한 것이다.

## 맺음말

디스크 전체에 대한 라이트 작업을 수행하는 DS.EXE, FR.EXE, SD.EXE를 수행하기 전에 꼭 DISKCOPY.EXE, BACKUP.EXE를 사용하여 디스크의 백업을 받아 놓아야 한다. 그렇지 않으면 불의의 사고(단전, 시스템의 폭주), 실수에 의하여 모든 내용을 잃어버린다.

끝으로 이 기사가 여러분께 도움이 되기를 바라며 여러분에 의해서 만들어진 유틸리티의 사용법 기사를 싣게 되기를 바라면서 기사를 마친다. PC

# SE-BUS

장경희/엘렉스컴퓨터

세계 PC시장의 양대산맥중 한 부분을 차지하고 있는 애플사의 16비트 기종인 매킨토시 SE을 살펴봄으로서, SE-BUS에 확장 카드를 인터페이스 시킬 수 있는 방법과 매킨토시에서만 사용하고 있는 독특한 버스 구조인 SE-BUS에 대해서 알아보자.

## 하드웨어 사양

매킨토시 SE 컴퓨터는 비디오 기능과, 사운드 기능이 마더보드상에 내장되어 있는 콤팩트한 설계로 되어 있으며, 그림 1에서와 같이 구성되어 있다.

●프로세서

MC68000
32비트 내부 아키텍처
16비트 외부 데이터 버스
24비트 외부 어드레스 버스
7.8336MHz 클럭 주파수

●램 메모리

1MB램(4MB까지 확장가능)
256B의 사용자 파라미터 메모리(불휘발성 램)

●플로피 디스크

800KB 양면 3.5″ 플로피 디스크 드라이브 한개 내장, 두개째는 선택으로서 내장형·외장형 모두 가능

●하드 디스크

20MB 내장 하드 디스크 선택으로 외장형 하드 디스크를 계속 연결 가능

●화면

9인치, 512×342 비트맵 디스플레이

●인터페이스

① 애플 데스크 탑 버스 코넥터 두개(마우스, 키보드 등 ADB방식의 디바이스 연결 가능)
② RS-232/RS-422 시리얼 포트 둘
③ 외부 플로피 디스크 인터페이스
④ SCSI 인터페이스 코넥터
⑤ 사운드 포트

*ADB는 애플사의 동기식 시리얼 데이터 전송 방법을 말함.

●전원 용량

매킨토시 SE의 전원은 선택 사양인 ADB 디바이스와 내장 하드 디스크 그리고 확장 카드를 지원하도록 설계되어 있다. 다음 표 1은 이 추가되는 모듈들에 대한 허용 전원 용량을 나타낸다.

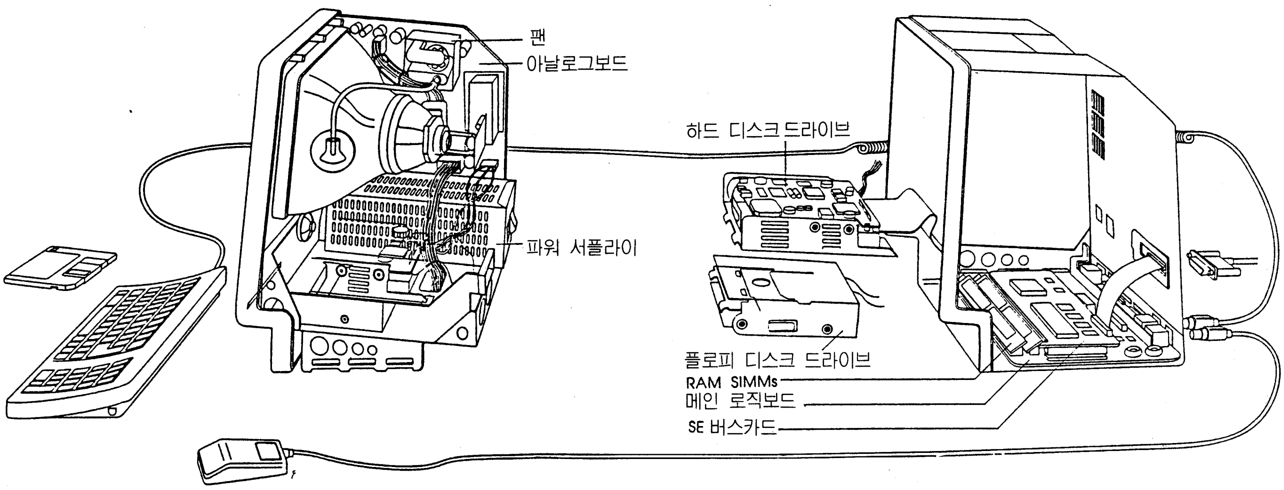
●기타

리튬전지 내장(캘린더 기능, 파라미터백업용)
확장 카드용 슬롯

## 하드웨어 아키텍처

SE-BUS와 확장 카드용 슬롯 사이

〔그림 1〕 매킨토시 SE의 구성



〔표 1〕 허용 전원용량

| 매킨토시 SE 디바이스 | －5V | －5V | ＋12V | －12V |
|---|---|---|---|---|
| 모든 ADB 디바이스들 | 0.5 | — | — | — |
| 내장 하드 디스크 | 1.5 | — | 0.9 | — |
| 확장 카드 | 1.5 | 0.1 | 0.15 | 0.1 |

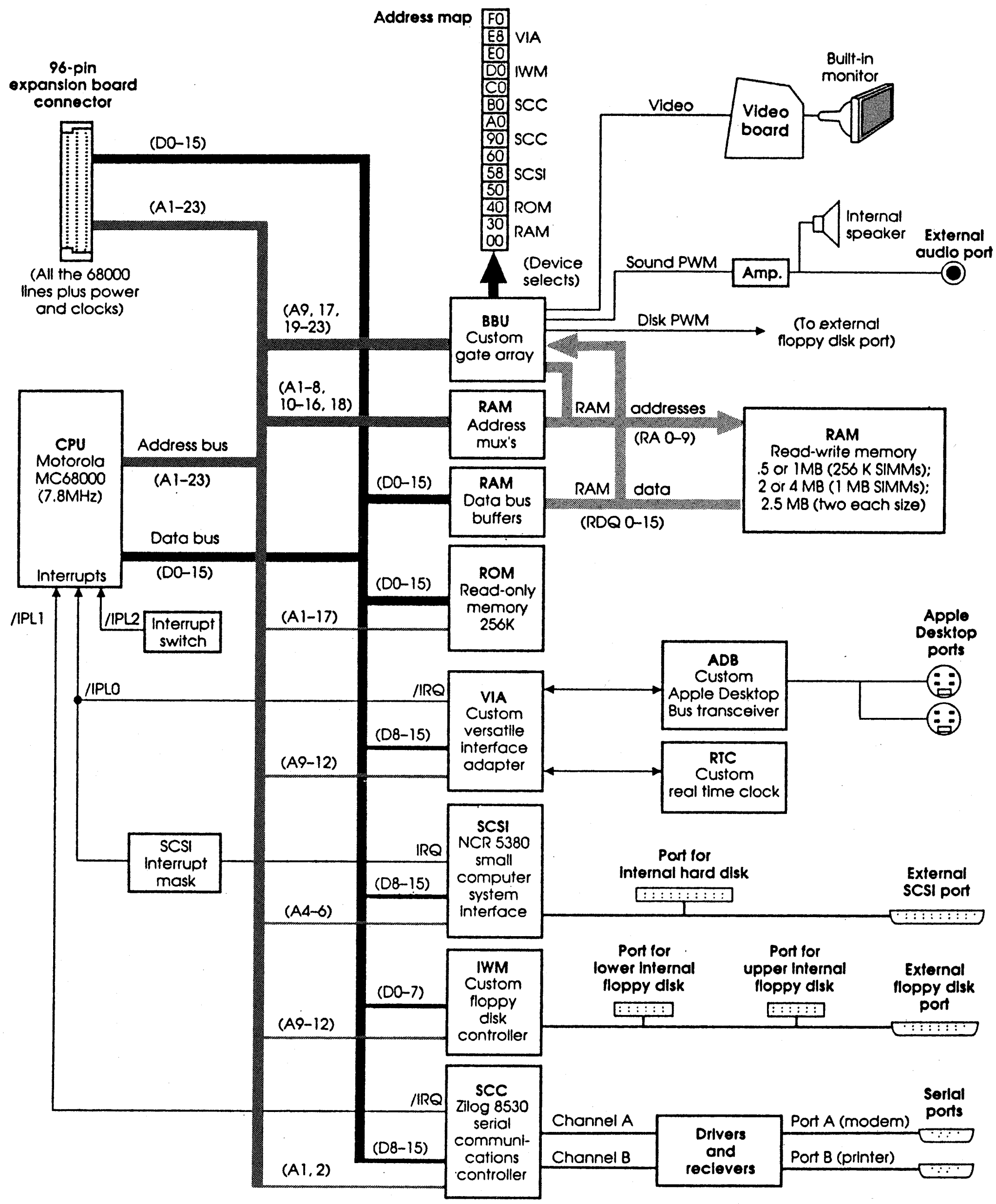
＊확장 카드의 경우 열발생을 고려하여 소비전력이 7와트를 초과하지 않도록 한다.　　＊단위 : 암페어

의 관계를 명확하게 보여주기 위하여 하드웨어의 구조와 동작원리를 설명하기로 한다.

매킨토시 SE는 7.8336MHz로 동작하는 16비트 마이크로 프로세서인 MC 68000을 중심으로 램과 롬 그리고 외부 디바이스들과의 통신을 가능하게 하는 다섯개의 I/O칩으로 구성되어 있다. 이러한 매킨토시 SE의 전체 구성을 블록 다이어그램으로 나타내면 그림 2와 같다.

## (1) NCR 5380 (SCSI 컨트롤러)

〔그림 2〕 매킨토시 SE 블록 다이어그램



하드 디스크 또는 SCSI 포트상에 있는 SCSI 디바이스들과의 고속 데이터 전송을 담당한다. 매킨토시에서는 전부 7개의 SCSI 디바이스들이 데이지 체인방식으로 연결될 수 있다.

### (2) Z8530(직렬통신 컨트롤러)

자이로그사의 통신용 칩으로서 동기 또는 비동기 방식의 직렬통신 기능을 담당한다.

매킨토시에서는 두개의 직렬포트가 있으며, 여기에서는 EIA-422 방식을 사용하고 있다.

### (3) IWM(Integrated Woz Machine)

애플의 창시자인 워즈니악의 이름을 따서 지은 애플사의 커스텀 칩으로서, 플로피 디스크 드라이브를 제어하는 플로피 컨트롤러 칩이다.

### (4) BBU(Bob Bailey Unit)

애플사의 커스텀 칩으로서 매킨토시 SE안에서 일어나는 다양한 로직 기능을 수행한다.

- 어드레스 디코딩 기능
- 비디오, 사운드 기능처리
- 클럭 시그널 생성
- 램 리플레쉬 기능
- 기타 각종 소자 사이의 인터페이스 역할을 수행함.

### (5) VIA

6522VIA 칩의 애플사 커스텀 버전으로서 ADB(Apple Desk Top Bus)와 RTC(Real-Time Clock)을 메인로직 보드와 인터페이스 시키는 일을 수행한다.

### (6) 데이터 버스

그림 2에서 보는 바와 같이 16개의 모든 데이터라인이 CPU, 롬, 램 데이터버퍼 그리고 SE-BUS에 연결되어 있다.

램 데이터 버스 버퍼는 램 데이터 버스에 메인 데이터 버스를 연결시키거나 또는 이 둘을 분리시키고, BBU가 램을 직접 액세스 할 수 있게 한다.

BBU는 비디오나 사운드 회로에 데이터를 제공하기 위하여 램을 직접 액세스하는 것이다.

시스템내에 있는 기타 다른 디바이스들은 데이터 버스의 상위 8비트(D15~D8) 또는 하위 8비트(D7~D0)에 연결된다.

### (7) 어드레스 버스

MC68000은 24비트 어드레스 버스 라인이 나와 있으며 총 $2^{24}$개(16MB)의 어드레스 공간을 제공한다.

램과 롬 그리고 VIA, SCSI, IWM, SCC 등을 선택하는 데도 메모리 어드레스라인을 사용하는 메모리-맵 방식을 사용한다.

## SE-BUS

매킨토시 SE-BUS의 주요한 특징은 매킨토시에서 CPU로 사용하고 있는 MC68000의 모든 신호에 직접 액세스 할 수 있는 기능을 제공하고 있는 96핀 DIN 코넥터에 있다.

즉 우리는 이 SE-BUS 코넥터를 통하여 각종 DMA 기능을 갖고 있거나, 코프로세서 등이 탑재되어 있는 새로운 주변기기들을 연결시킬 수 있다.

확장 카드나 주변 I/O 디바이스들은 매킨토시 어드레스 영역의 일부분을 차지하게 되고, 매킨토시는 이 카드를 자신의 I/O 디바이스중의 어느 하나로 인식하고 액세스 하게 된다.

### (1) 버스 코넥터 핀 배치도

그림 3은 매킨토시 SE의 96핀 DIN 코넥터에 있는 각 핀의 신호를 나타내고 있으며, 그 각각의 기능은 다음과 같다.

### (2) 각 핀의 전기적특성

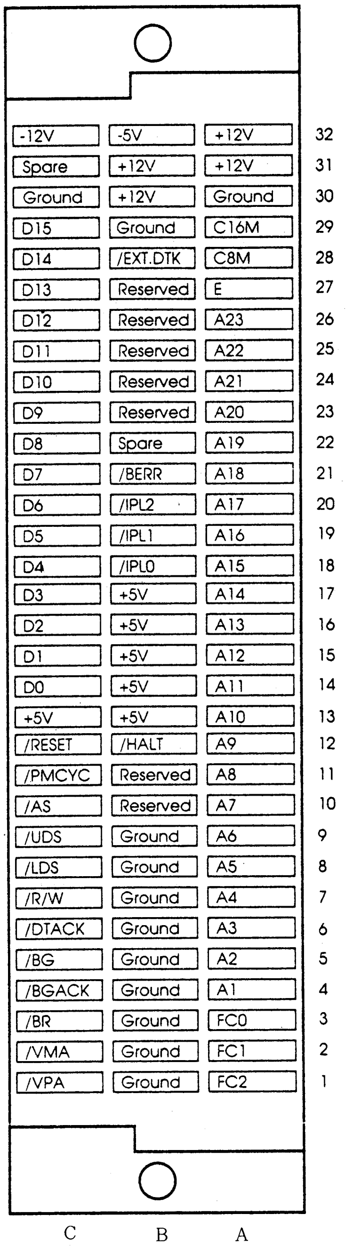
다음의 표 2에는 각 핀의 신호가 제공하는 부하능력과 구동능력을 나타낸다.

### (3) 각 핀의 기능 설명

**●FC0~FC2**

이 세핀의 출력은 현재 수행되고 있는 프로세서의 메모리 액세스 상태와 인터럽트 접수 상태를 나타내는 기능코드로, 외부 디바이스에 대해 68000의 동작상황을 전달한다. 이 FC0~FC2에

〔그림 3〕 96핀 확장 코넥터



〔표 2〕 96핀의 전기적 특성

| 코넥터 열 | 코넥터 핀 | 신호이름 | 신호정의 | 입·출력 | 기타 |
|---|---|---|---|---|---|
| A | 1 | FC2 | Function code 2 | Output (Input) | Drive: 40μA/.4mA, 30pF (Load: 100μA/100μA, 50pF) |
| A | 2 | FC1 | Function code 1 | Output (Input) | Drive: 40μA/.4mA, 30pF (Load: 100μA/100μA, 50pF) |
| A | 3 | FC0 | Function code 0 | Output (Input) | Drive: 40μA/.4mA, 30pF (Load: 100μA/100μA, 50pF) |
| A | 4–26 | A1–23 | Address 1–23 | In/Out | Load: 250μA/1mA, 100pF Drive: 40μA/.4mA, 30pF |
| A | 27 | E | E (enable) clock | Output | Drive: 40μA/.4mA, 30pF |
| A | 28 | C8M | 7.8336 MHz MC68000 clock | Output | Drive: 20μA/.2mA, 20pF |
| A | 29 | C16M | 15.6672 MHz gate array and IWM clock | Output | Drive: 20μA/.2mA, 20pF |
| A | 30 | GND | Logic ground | | |
| A | 31 | +12V | +12 volts | Output | Drive: 150mA total, from all +12V pins |
| A | 32 | +12V | +12 volts | Output | (see the section "Power Budget") |
| B | 1–9 | GND | Logic ground | | |
| B | 10 | Reserved | For future Apple use; do not connect | | |
| B | 11 | Reserved | For future Apple use; do not connect | | |
| B | 12 | /HALT | MC68000 Halt | In/Out | Load: 300μA/6mA, 50pF Drive: 0μA/0μA (Connected to /RESET, pin C-12) |
| B | 13–17 | +5V | +5 volts | Output | Drive: 1.5 A total, from all +5V pins (see the section "Power Budget") |
| B | 18 | /IPL0 | Interrupt level 0 (VIA, SCSI.IRQ) | In/Out | Load: 100μA/2mA, 50pF Drive: 40μA/.4mA, 30pF (Open collector; 3.3K ohm pullup) |
| B | 19 | /IPL1 | Interrupt level 1 (SCC) | In/Out | Load: 100μA/2mA, 50pF Drive: 40μA/.4mA, 30pF (Open collector; 3.3K ohm pullup) |
| B | 20 | /IPL2 | Interrupt level 2 (NMI switch) | In/Out | Load: 100μA/2mA, 50pF Drive: 40μA/.4mA, 30pF (Open collector; 3.3K ohm pullup) |
| B | 21 | /BERR | Bus error | In/Out | Load: 100μA/2mA, 50pF Drive: 40μA/.4mA, 30pF (Open collector; 3.3K ohm pullup) |
| B | 22 | Spare | Not connected | | |
| B | 23–27 | Reserved | For future Apple use; do not connect | | |
| B | 28 | /EXT.DTK | External /DTACK (tri-states main board's /DTACK) | Input | Load: 100μA/2mA, 50pF (3.3K ohm pullup) |
| B | 29 | GND | Logic ground | | |
| B | 30 | +12V | +12 volts | Output | Drive: 150mA total, from all +12V pins |
| B | 31 | +12V | +12 volts | Output | (see the section "Power Budget") |
| B | 32 | -5V | -5 volts | Output | Drive: 100mA |
| C | 1 | /VPA | Valid periph. address | Output | Drive: 40μA/.4mA, 30pF |
| C | 2 | /VMA | Valid memory address | Output (Input) | Drive: 40μA/.4mA, 30pF (Load: 100μA/100μA, 50pF) |
| C | 3 | /BR | Bus request | Input | Load: 100μA/2mA, 50pF (3.3K ohm pullup) |

나타난 정보는 AS신호가 액티브 상태일때 유효하다.

| FC2 | FC1 | FC0 | 기 능 |
|---|---|---|---|
| 0 | 0 | 0 | --- |
| 0 | 0 | 1 | 유저 데이터 |
| 0 | 1 | 0 | 유저 프로그램 |
| 0 | 1 | 1 | --- |
| 1 | 0 | 0 | --- |
| 1 | 0 | 1 | 수퍼바이저 데이터 |
| 1 | 1 | 0 | 수퍼바이저 프로그램 |
| 1 | 1 | 1 | 인터럽트 액크놀로지 |

### ● A1~A23

A1에서 A23까지의 어드레스 버스는 23비트의 단 방향성 3스테이트 버스로서 16M바이트의 데이터를 직접 어드레스 할 수 있다.

인터럽트 사이클 동안에는 어드레스 라인 A1, A2 그리고 A3는 어느 레벨의 인터럽트가 서비스되고 있는지를 나타낸다.

### ● E

이 출력 신호는 MC68000 주변 디바이스에 적용되는 표준 인에이블 신호이다. 이 신호의 주기는 MC68000클럭의 10배이고 버스의 상태하고는 무관하게 동작하는 클럭신호이다.

### ● C8M

마이크로 프로세서 클럭으로 C16M을 2분주하여 구해진다.

7.8336MHz =C16M/2

### ● C16M

게이트 어레이 클럭(BBU)=15.6672 MHz

### ● /HALT

이 쌍방향 신호는 외부에서 홀트 신호를 접수하면, 마이크로 프로세서는 동작중인 버스 사이클을 완료한 시점에서 정지한다. 그러면 모든 콘트롤 신호는 활동을 중지하고, 모든 3-스테이트라인은 하이 임피던스 상태로 간다. 또 마이크로 프로세서가 더블버스 폴트 상태에서 명령 실행을 정지했을 때는, 이 홀트라인이 외부 디바이스에게

| 코넥터 열 | 코넥터 핀 | 신호이름 | 신호처리 | 입·출력 | 기타 |
|---|---|---|---|---|---|
| C | 4 | /BGACK | Bus grant acknowledge | Input | Load: 100µA/2mA, 50pF (3.3K ohm pullup) |
| C | 5 | /BG | Bus grant | Output | Drive: 40µA/.4mA, 30pF |
| C | 6 | /DTACK | Data transfer acknowledge | In/Out | Load: 100µA/2mA, 50pF Drive: 40µA/.4mA, 30pF (3.3K ohm pullup, /EXT.DTK low, tri-states main board's /DTACK) |
| C | 7 | R/W | Read/write | Output (Input) | Drive: 40µA/.4mA, 30pF (Load: 200µA/2mA, 50pF) |
| C | 8 | /LDS | Lower data strobe | Output (Input) | Drive: 40µA/.4mA, 30pF (Load: 100µA/1mA, 50pF) |
| C | 9 | /UDS | Upper data strobe | Output (Input) | Drive: 40µA/.4mA, 30pF (Load: 100µA/1mA, 50pF) |
| C | 10 | /AS | Address strobe | Output (Input) | Drive: 40µA/.4mA, 30pF (Load: 200µA/3.2mA, 50pF; 3.3K ohm pullup) |
| C | 11 | /PMCYC | Processor memory cycle | Output | Drive: 40µA/.4mA, 30pF (High during video access to RAM) |
| C | 12 | /RESET | System reset | In/Out | Load: 300µA/6mA, 50pF Drive: 40µA/.4mA, 30pF (Open collector; 1K ohm pullup) (Connected to /HALT, pin B-12) |
| C | 13 | +5V | +5 volts | Output | Drive: 1.5A total, from all +5V pins (see the section "Power Budget") |
| C | 14-29 | D0-15 | Data bus bits 0-15 | In/Out | Load: 250µA/1mA, 100pF Drive: 40µA/.4mA, 30pF |
| C | 30 | GND | Logic ground | | |
| C | 31 | Spare | Not connected | | |
| C | 32 | -12V | -12 volts | Output | Drive: 100mA |

마이크로 프로세서가 정지 하였음을 알린다.

**●/IPL0~/IPL2**

3개의 입력신호는 외부에서 인터럽트를 요구하는 디바이스의 우선레벨을 정하는 것이다. IPL0가 최하위 비트이고 IPL2가 최상위 비트이며 레벨 "1"이 가장 하위이고 레벨 "7"이 최우선 레벨을 나타낸다.

| 인터럽트 마스크 | I2 | I1 | I0 | 인터럽트 우선도 | 인터럽트 내용 | 벡터 번호 | 벡터 어드레스 |
|---|---|---|---|---|---|---|---|
| 7 | 1 | 1 | 1 | 1위 | 레벨 7 인터럽트 오토벡터 | 31 | 07C |
| 6 | 1 | 1 | 0 | 2위 | 레벨 6 인터럽트 오토벡터 | 30 | 078 |
| 5 | 1 | 0 | 1 | 3위 | 레벨 5 인터럽트 오토벡터 | 29 | 074 |
| 4 | 1 | 0 | 0 | 4위 | 레벨 4 인터럽트 오토벡터 | 28 | 070 |
| 3 | 0 | 1 | 1 | 5위 | 레벨 3 인터럽트 오토벡터 | 27 | 06C |
| 2 | 0 | 1 | 0 | 6위 | 레벨 2 인터럽트 오토벡터 | 26 | 068 |
| 1 | 0 | 0 | 1 | 7위 | 레벨 1 인터럽트 오토벡터 | 25 | 064 |

**●/BERR**

이 신호는 프로세서에게 현재 수행된 버스사이클에 문제가 있음을 알린다. 즉 SCSI 액세스 타임아웃이 되면 매킨토시에 있는 게이트 어레이(BBU)에 의해 발생한다. 실제로 /BERR은 AS가 /250ms 이상 로우상태로 남아 있으면 만들어진다.

**●EXT.DTK**

게이트 어레이의 DTACK 출력을 하이 임피던스 상태로 하는 신호이며, 이후 확장 카드가 /DTACK 신호를 만들어내게 된다.

/DTACK는 데이터 전송이 완료되었음을 프로세서에게 알리는 신호이다.

**●/VPA**

VPA영역($ E0 0000~$FF FFFF)을 액세스 하는 경우 /AS와 일치하도록 게이트 어레이에 의해 제공된다.

**●/VMA**

이 출력은 M68000 주변 디바이스에게 어드레스 버스에 유효한 어드레스가 있고, 이 프로세서는 인에이블 신호에 동기화되어 있음을 알린다.

이 신호는 VPA 입력에만 응답한다.

**●/BR**

이 입력은 버스 마스터가 되고자 하는 모든 디바이스가 같이 결선(Wired-OR)되어 제공할 수 있으며, 프로세서에게 버스를 사용하겠다고 알린다.

**●R/W**

이 신호는 데이터 버스 전송상태가 리드 사이클 또는 라이트 사이클 인지를 정의한다. 이 R/W 신호는 또한 데이터 스트로브 신호와 연결하여 동작된다.

**●/LDS, /UDS**

이 신호들은 R/W와 함께 데이터 버스상의 데이터 흐름을 제어한다(표 3).

**●/AS**

어드레스 버스에 유효 어드레스가 있음을 알린다.

**●/PMCYC**

프로세서가 램을 액세스 할 수 있을 경우 로우로 되며, 비디오 액세스 중에는 하이 상태가 된다.

램을 액세스 하고자 할때 게이트 어레이와 동기 시키는데 사용된다.

**●/RESET**

양방향 신호로서 외부의 리세트 신호에 응답하여 프로세서를 리세트 시킨다. RESET라는 명령어에 의해 내부적으로 만들어진 리세트 신호는 모든 외부 디바이스를 리세트 시키나, 프로

〔표 3〕 UDS, LDS를 이용한 데이터 제어

| UDS | LDS | R/W | D8~D15 | D0~D7 |
|---|---|---|---|---|
| High | High | — | 무효 데이터 | 무효 데이터 |
| Low | Low | High | 유효 데이터 | 유효 데이터 |
| High | Low | High | 무효 데이터 | 유효 데이터 |
| Low | High | High | 유효 데이터 | 무효 데이터 |
| Low | Low | Low | 유효 데이터 | 유효 데이터 |
| High | Low | Low | 유효 데이터 | 유효 데이터 |
| Low | High | Low | 유효 데이터 | 유효 데이터 |

세서 자신의 상태는 변화되지 않는다.

## 확장 카드에서의 매킨토시 액세스

### (1) I/O 디바이스 액세스

대부분의 I/O 디바이스를 확장 카드에서 액세스 하고자 할때 액세스 타이밍은 확장 카드에 있는 코프로세서가 제어하게 된다. 이 코프로세서가 어드레스 버스에다 특정 디바이스의 어드레스를 내보내고 어드레스 스트로브(/AS) 신호를 내보낸다. 그러면 커스텀 칩인 BBU가 올바른 디바이스를 선택하는 신호를 내보내고 /DTACK을 발생시킨다.

만일 카드를 설계하여 /DTACK 라인에 자기가 원하는 신호를 공급하고자 하면, /EXT.DTK 라인을 로우상태로 함으로서 BBU 칩의 /DATCK 출력을 트라이스테이트 상태로 해야 한다. 또한 액세스 하고자 하는 영역이 $E000000~$FFFFFF에 있다면(예로써 VIA 칩을 액세스 하는 경우), BBU 칩은 /DATCK 신호 대신에 /VPA을 제공한다. 그러면 정상적인 동작의 경우에 매킨토시 SE 로직보드에 있는 MC68000은 VIA 칩 선택 신호인 /VMA을 제공함으로서 /VPA에 응답한다.

### (2) 램 액세스

확장 카드에 있는 코프로세서가 매킨토시 SE 로직보드에 있는 램을 액세스하는 경우, 액세스 타이밍은 I/O 디바이스를 액세스하는 경우보다 더욱 제한을 받게 된다.

설사 코프로세서가 확장 카드에 있는 자신 전용의 램 메모리를 갖고 있다 하더라도, 화면에 나타난 정보를 개정하기 위해서는 매킨토시 SE 램을 액세스할 필요가 있다. 왜냐하면 매킨토시 화면에 나타난 정보는 매킨토시 시스템 램에서 취하기 때문이다. 그러므로 확장 카드를 설계하는 경우 매킨토시 비디오회로가 정기적으로 램 메모리를 액세스하는 것을 고려하여 잘 동기화시킬 수 있도록 한다.

화면을 주사하는 활동 기간중에 비디오회로는 매 4번의 가능한 램 액세스 타이밍중 한번을 사용한다. 이 비디오 액세스는 시스템 내에서 일어나고 있는 다른 활동에 관계없이 고정된 시간마다 발생한다.

만일 비디오 회로가 램을 액세스할 때 코프로세서가 매킨토시 램 액세스를 시작하면 이 코프로세서의 액세스는 다음 램 액세스 타임까지 홀드되어 연기된다(DTACK가 제공되지 않음).

**●비디오/사운드 사이클**

비디오 카운트 회로의 카운트 결과로서 주기적으로 발생한다. 비디오 사이클은 램에 있는 비디오 버퍼에서 두 워드의 데이터를 읽어, 게이트 어레이(BBU)의 비디오 시프트 레지스터에 저장한다.

사운드 사이클은 비디오 사이클과 유사하나 램에 있는 사운드/디스크 스피드 버퍼로 부터 단일 워드만을 읽어, 게이트 어레이의 사운드/디스크 스피드 카운터에 저장한다.

**●프로세서 사이클**

이것은 비디오/사운드 사이클중이 아니면 언제나 램을 액세스할 수 있는 사이클이다. 만일 프로세서가 비디오/사운드 사이클동안에 램을 액세스하려고 하면 /DTACK가 공급되지 않아 비디오/사운드 사이클이 완료될때까지 지체된다. 그러나 비디오/사운드 사이클일지라도 램 이외의 다른 디바이스들은 액세스 할 수 있다.

BBU 칩은 확장 카드에 이 프로세서가 임의의 시간에 램 액세스를 시작할 수 없게 한다. 정상 상태의 경우 BBU는 어떤 프로세서이든 그것이 MC68000과 같이 행동할 것을 기대하고 있다. 다시 말하면 매킨토시 SE에서 MC68000은 /DTACK을 S4 상태에서 받아들임으로서 자동적으로 BBU에 있는 스테이트 머신에 동기된다.

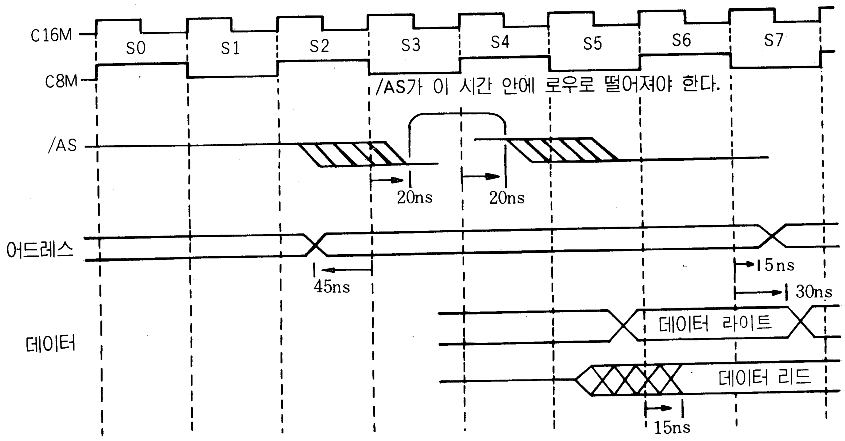
확장 카드는 /PMCYC라는 신호를 모니터 함으로서 BBU에 있는 스테이트 머신에 자신을 동기시킬 수 있게 된다. /PMCYC는 항상 8스테이트 중 S1에서 로우로 떨어진다. C16M에 의해 트리거되는 회로를 구성하면 /PMCYC의 폴링 에지를 검출할 수 있을 것이다.

다음 **그림 4**는 확장 카드에서 램을 리드하거나 라이트하는데 필요한 타이밍 챠트이다.

그림 4에서 보면 /AS가 로우로 떨어지기 전에 필요한 최소한의 어드레스 셋업 타임은 15ns이며 S3가 시작하기 전에 필요한 최소한의 어드레스 셋업 타임은 45ns로 나타나 있다.

어드레스는 S7의 첫 5ns동안 유효상태를 계속 유지하여야 한다. 또한 /AS가 로우로 떨어지는 것은 S3가 시작되면서 20ns가 지나서는 안된다. 만일 /AS가 그 시간까지 로우로 떨어지지 않으면 S4의 첫 20ns가 지나기 전에 로우로 떨어져서도 안된다. 이럴 경우 데이터는 다음 램 액세스타임에 리드되

〔그림 4〕 확장카드에서의 램 액세스




거나 라이트되게 된다.

/DTACK는 /AS가 상승한 후 바로 뒤를 따르는 첫 홀수 S—스테이트에서 최대 25ns까지 상승해 있어야 한다. 왜냐하면 램에 라이트 될 데이터는 S6의 시작에서 부터 S7의 첫 30ns까지만 유효하기 때문이다.

*그림에서 C8M 클럭은 MC68000의 스테이트와 관련하여 보여주기 위한 것이며 실제로는 C16M에 비하여 30ns 정도 지연되어 있다.

## (3) 타이밍 차트

다음 **그림 5**는 비디오 회로와 MC68000이 램을 액세스하는 타이밍 차트이다. 여기에서 BBU 칩은 C16M에 의해 클럭을 받아 16스테이트(S0～SF)을 만들어내는 내부 스테이트 머신을 갖고 있다. 이 스테이트 머신은 16스테이트를 연속적으로 카운트하여 나아가며 시스템에 있는 다른 상태에 의해 영향을 받지 않는다.

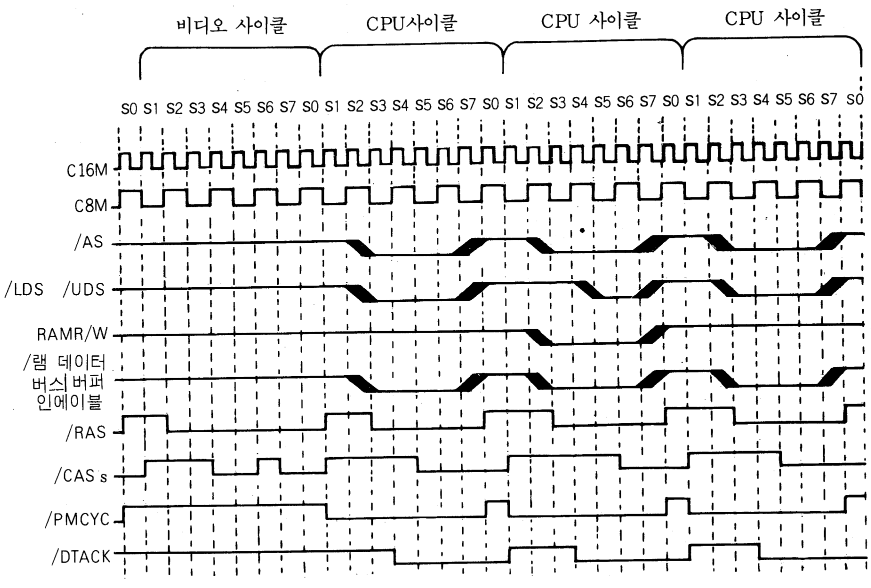
매킨토시 SE에서는 비디오/사운드 사이클과 프로세서 사이클이라는 2가지 타입의 기본적인 램 액세스 사이클이 있다. 이중 어느 타입의 램 액세스 사이클이라도 8개의 스테이트 머신 상태를 가지고 있다. 비디오/사운드 사이클은 항상 S8에서 SF까지 있고, 프로세서 사이클은 S0에서 S7 또는 S8에서 SF 상태에 있을 수 있다. 그러나 단순하게 이야기하기 위해 여기서는 어떤 타입의 램 액세스 사이클이든 S0에서 S7까지 번호를 주기로 한다.

C16M 신호는 매킨토시 로직에 있는 BBU가 제공하는 15.6672MHz 클럭 이다. C8M 신호는 이 신호를 2분주하여 구해지며 사이클의 시작은 스테이트 0(S0)이다. S2에서는 /AS, /LDS, /UDS 등이 어서트 된다. 어드레스가 출력이 되고 나서 /AS, /LDS, /VDS가 어서트 되기까지의 시간은 약 50ns 이다. 그러므로 외부 회로의 어드레스 디코더는 이 시간동안에 자신의 출력을 안정시키지 않으면 안된다.

S3, S4에서는 이 상태를 유지하고, S4의 종료시에 CPU는 /DATCK 신호를 샘플한다. S5, S6에서 출력신호의 변화가 없고 S6의 종료시에 MC68000은 메모리로 부터 나와 데이터 버스에 실려있던 데이터를 읽어 들인다. 따라서 데이터는 S7의 종료보다 데이터 셋업시간(15ns)앞에 확정되어 있어야 한다

S7에서 CPU는 /AS, /LDS, /UDS를 니케이트하여 사이클을 종료시킨다.

〔그림 5〕 비디오와 MC68000의 램 액세스

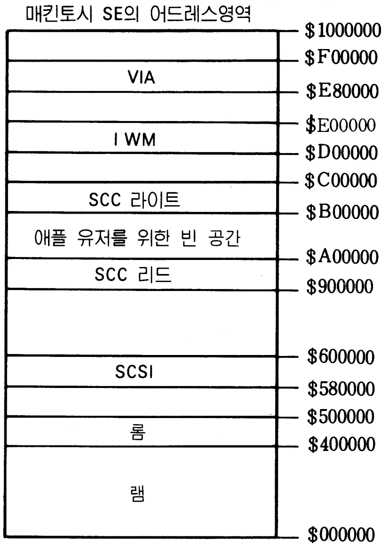



# 어드레스 맵

다음 **그림 6**의 어드레스 맵은 현재 매킨토시 SE가 사용하고 있는 전체 어드레스 영역을 나타낸다. 매킨토시 SE 하드웨어가 사용하고 있지 않은 어떤 어드레스 영역도 확장 카드에서 사용할 수 있으나 다음의 제한이 따른다.

$000000에서 $DFFFFF까지의 어드레스 공간에 대한 액세스에 대하여, BBU 칩은 S2 단계에서 /AS를 제공하고 그뒤를 이어 S4 단계에서 /DTACK

〔그림 6〕 매킨토시 SE의 어드레스 맵




를 제공한다.

만일 /AS가 S3 이후에 로우로 떨어지면 /DTACK 다음 액세스 사이클의 S0에서 제공된다. 물론 다음 사이클의 S4까지 기다리게 되는 램 액세스의 경우는 제외가 된다. 이 영역은 고속 액세스를 요하는 경우 적합하다.

$E00000에서 $FFFFFF 영역에 대한 액세스를 하는 경우, BBU 칩은 /AS를 제공하고 바로 뒤이어 NPA신호를 내보낸다. 그리고 MC68000은 E클럭에 의해 동기된 /VMA 신호를 제공한다. 이 영역은 6800 스타일의 동기식 주변기기 등 느린 속도의 액세스시 적합하도록 설계된 것이다.

매킨토시 SE램은 항상 $000000에서 $3FFFFF까지의 전 영역을 차지한다.

매킨토시 SE는 한개 이상의 확장 카드 연결을 지원하지는 않는다. 즉 다중 프로세서들을 조정하거나, 같은 어드레스 영역을 사용하는 카드 등을 조정하는 기능이 없다.

## EMI 가이드라인

매킨토시 SE는 FCC의 EMI 요구기준에 맞게 설계되어 있다. 그러므로 매킨토시 SE에 추가되는 확장 카드를 설치할 경우 이 기준을 벗어나지 않도록 주의하여야 한다. EMI문제는 신호 전송로를 재배치하거나 각 버스를 적절히 터미네이션 함으로서 그리고 필터링 방법 등을 이용하여 해결할 수 있다. 확장 카드를 설계할 경우는 다음 사항에 주의하여 설계하여야 한다.

- 최소한 4층 기판을 사용하여 EMI 방출을 감소시킨다.
- 고속의 신호는 버퍼를 하고, 저속의 회로와 분리시킨다.
- 가능한 한 96핀 코넥터에 가까운 곳에서 버퍼를 실시하고, 최대 30PF의 정전용량을 갖는 한개의 LS부하 정도로 제한한다.
- 내부의 케이블 연결은 가능한 짧게 하고 기존의 매킨토시 부품과 정전결합이나 전자 결합을 일으키지 않도록 배치한다.
- 고속의 신호를 전달하는 도체는 저속의 신호를 전달하는 도체와 함께 묶여서는 안된다.
- 어떤 경우에도 실드된 선 또는 트위스트디 페어선을 사용하여야 된다.
- 전원과 연결되는 곳에 적절한 전원 필터를 설치하고 고주파 결합을 피하도록 한다. PC

# 애플컴퓨터를 사용한
# 조명제어장치

남영택, 채현우/한국항공대학 전자과

필자는 여러가지 조명기구를 다루다 보니 컴퓨터를 이용하여 전력을 제어하는 방법을 생각해 보았다. 온/오프 제어는 간단하게 구성될 수 있었으나 전구의 밝기를 조절하는 경우는 직류를 제어하는 것과 교류를 제어하는 것으로 나뉘어 있으므로 생각처럼 그리 간단하게 되지는 않았다.

보통 가정용 전원의 교류전력을 제어하기 위해서는 특수반도체 소자중에 하나인 트라이액을 사용하여 위상제어를 하는 방식이 가장 적합하다고 생각하여 이 장치를 제작해 보았다.

## 개　　요

전력제어 방식에는 전압제어 방식과 위상제어 방식으로 나누어진다. 전자는 저항성분을 이용한 것이고 후자는 스위칭 작용을 이용한 것으로 먼저 두가지 방식을 소개하고 어떤 방식이 손실면에서 우수하고 컴퓨터를 이용하는데 용이한가에 대해서 밝혀 보기로 하자.

### ●전압제어방식

전원과 직렬로 가변저항이나 트랜지스터를 넣어 부하에 걸리는 전압과 전류를 조절하는 것으로 교류제어에는 트랜지스터의 사용이 어렵고 직류에 있어서 가능하다.

그림 1(a)에서와 같이 500W의 전구를 보자. 어느 곳에서도 전력손실이 없고 부하전구에 500W의 전력이 소비될 뿐이다. 그림 1(b)에서 전구의 저항치와 같은 저항을 달면 전류는 2.5A가 흐르고 전구와 저항이 똑같이 125W의 전력이 소비된다. 밝기를 조절하기 위해 이만한 전력이 손실된다. 저항대신 트랜지스터를 이용하여 전력을 제어하려면 부하와 같은 전력을 발생시켜도 좋은 트랜지스터를 써야 되며 손실전력도 큰것이 필요하다.

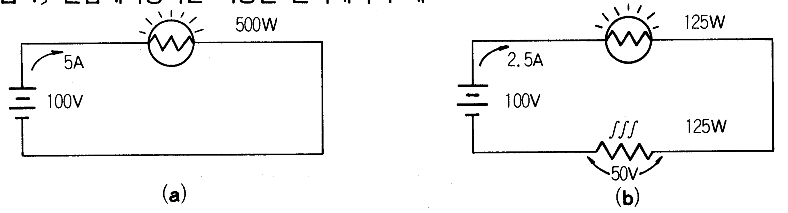
### ●위상제어방식

스위칭동작 소자인 다이리스터를 이용하는 방법으로 여기서는 트라이액을 이용하였다. 그림 2에 트라이액의 구조와 기호를 그림 3에 전압-전류특성을 보인다.

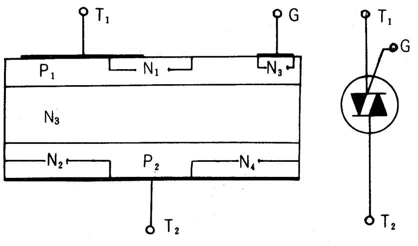
이 소자의 특징은 양방향성 소자로 게이트의 극성에 무관하게 작은 전류를 흘려 보내면 순차적으로 ON상태로 되는 것으로 전원이 OFF되기전 까지는 ON상태를 유지한다. 트라이액을 이용하여 전력을 제어하는 경우는 그림 4(a)와 같이 장치하고 게이트라는 전극에 펄스를 주면, 트라이액은 ON이 되고, 트라이액의 양단 전압이 제로가 되면 그 순간부터 트라이액은 OFF가 되므로 그림 4(b)와 같은 교류파형일때, 그 아래와 같은 펄스를 주면 웨이브의 후반에서만 전류가 흘러 이 부분이 전력이 된다.

교류 60Hz는 부하에 전구를 사용하는 경우에 그림 4에서와 같은 점멸을 120번 하는 것과 같지만 눈으로 식별할 수 없는 속도이기 때문에, 보기에는 밝기가 한결같이 약해진 것이라고 밖에는 느끼지 못한다. 펄스를 주는 시간(위상)을 빨리하면 흐르는 시간이 더해져서 전구가 밝아지고, 느리게 하면

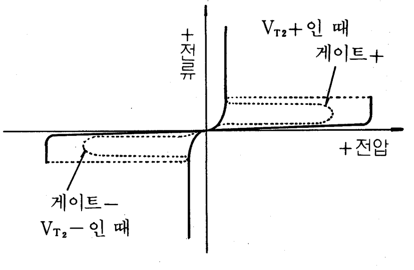
〔그림 1〕 전압제어방식을 이용한 전력제어의 예



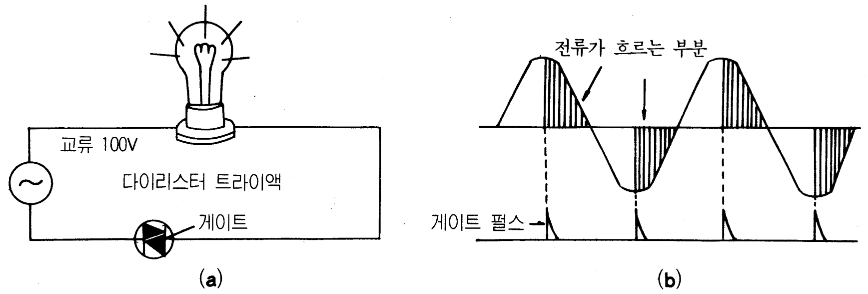

〔그림 2〕 트라이액의 구조와 심볼




〔그림 3〕 트라이액의 특성



〔그림 4〕 다이리스터에 의한 전류제어




흐르는 시간이 줄어, 전구는 어두워진다. 이와 같이 다이리스터에 의한 전력제어는, 저항이 아니고 스위치에 의한 것으로 전력손실은 트라이액 양단의 전압강하가 0.6V 정도로 5A의 전류를 흘린다 하여도 5A×0.6V=3W 정도의 작은 전력이 손실된다. 컴퓨터를 이용하여 제어하는 경우는 그림 4(b) 아래 그림의 게이트 펄스위상각만 제어해 주면 된다.

## 블록 다이어그램

전체적인 구성도는 그림 5에 보인다.

### ●CPU

0~255의 8비트 데이터를 발생하고 조명기구를 지정한다. 여기서는 6502 CPU를 사용한 애플컴퓨터를 이용했다.

### ●인터페이스

68계열의 입출력 LSI 6821PIA를 사용하여 컴퓨터의 데이터와 조명기구 설정어드레스를 처리하여 래치I으로 보내준다.

### ●래치 및 비교회로

이 부분은 각 조명기구에 하나씩 할당된다. 어드레스신호와 8비트의 데이터는 래치 I 으로 들어가 어드레스신호가 래치 I 을 인에이블 상태로 하여 8비트의 데이터와 트리거-카운터회로에서 나온 8비트의 데이터를 비교하여 데이터가 같을 경우 래치II를 인에이블하여 트리거회로의 트리거된 파형을 조명제어 회로로 보내준다.

### ●트리거-카운터회로

60Hz 전원과 동기된 구형파를 만들고 이 파형을 래치II로 보내준다. 또 이 구형파를 이용하여 리세트 펄스를 만들어 카운터와 클럭이 1/120초마다 클리어될 수 있게 한다. 카운터는 1/120초마다 0~255의 수를 세고 이 8비트 데이터를 비교회로로 보내 래치 I 의 8비트 데이터와 비교하게 된다.

### ●클럭-분주회로

3.932160MHz의 수정발진을 하고 $1/2^7$분주하여 30.72kHz의 구형파를 카운터로 보내 카운터가 수를 셀 수 있게 한다.

### ●조명제어부

비교회로에서 나온 신호는 래치II를 인에이블하여 래치II에 보내진 트리거된 파형으로 트라이액을 구동한다.

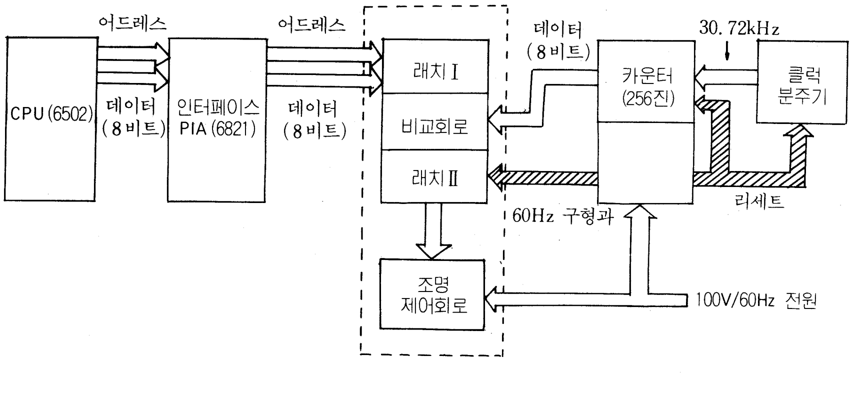
## 인터페이스

여러가지의 조명을 제어할 경우 각각의 조명기구를 선택해줄 어드레스와 각각의 밝기를 지정해줄 데이터가 필요하다. 본 제작에서는 이 어드레스와 데이터를 컴퓨터로부터 출력하기 위해 PIA 6821을 사용하였다. 이 PIA는 6800계열의 주변기기로 설계되었으나 우리가 사용한 애플 CPU인 6502와 유사한 특성을 가지고 있기 때문에 별 문제없이 사용할 수 있다.

이 PIA는 입출력용으로 A포트와 B포트를 가지고 있어 입출력 비트수를 자유로이 정할 수 있다. 이번 제작에서는 A포트, B포트 모두 출력용으로 정하고 A포트에서는 어드레스를, B포트에서는 데이터를 출력하도록 설계하였다. 이 PIA는 처음 프로그램 상에서 각각의 입출력 비트를 입력으로할 것인지 아니면 출력으로할 것인지를 정해주어야 하는데 이것을 초기화 시킨다고 말하고 있다. PIA6821을 사용함에 있어서는 이 초기화 시켜주는 방법과 컴퓨터의 슬롯 단자와 이 PIA단자와의 연결만 잘해주면 문제될 것은 없으리라 생각한다.

PIA에는 A포트, B포트용으로 CR A, B(Contral Register A, B), DRA, B (Data direction Register A, B), PR A, B(Peripheral data Register A, B)라는 6개의 레지스터가 있다. 하지만

〔그림 5〕 조명제어회로의 전체 구성도 (그림의 점선부분은 각 조명기구에 1개씩 할당)



이 6개의 레지스터를 선택해줄 단자는 RS0와 RS1 2개밖에 없다.

A포트와 B포트는 서로 같은 방식을 취하므로 우선 A포트만 가지고 생각해 보자. CRA에는 8비트의 데이터가 기억되는데 이 8개의 비트중 3번째비트(비트2)는 DRA와 PRA중 하나를 선택하도록 되어 있다. 즉 비트2가 "0"이면 DRA를, "1"이면 PRA를 선택하도록 되어있다. 다시 말해서 DRA와 PRA는 RS0, RS1에 의해 선택되는 값은 같지만 CRA의 3번째 비트값에 의해 먼저 DRA나 PRA중 하나가 선택되는 것이다.

그리고 이 DRA의 8비트 기억장소에는 PRA의 각 비트가 입력용인지 아니면 출력용인지를 기억하고 있다. 즉 DRA의 n비트에 "0"이 기록되어 있으면 PRA의 n비트는 입력용으로, DRA의 n비트에 "1"이 기록되어 있으면 PRA의 n비트는 출력용으로 지정되는 것이다. 나중에 초기화시켜주는 프로그램을 보면 알겠지만, 먼저 CRA의 2비트에 "0"을 넣어 DRA를 선택하고 입력용으로 사용하고자 하는 비트는 "0"을, 출력용으로 사용하고자 하는 비트는 "1"을 기억시키고 나서 다시 CRA의 2비트에 "1"을 넣어주면 초기화는 끝나는 것이다.

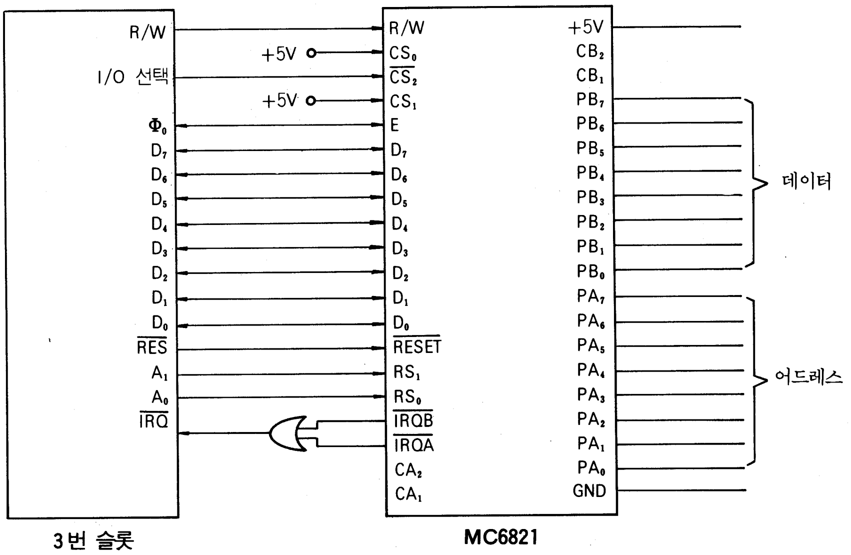
처음에는 $RS0 = A_0$, $RS1 = A_1$에 연결되어 있고 인터페이스 카드는 애플의 3번 슬롯에 꽂아져 있으므로 PIA의 A포트는 C300번지로 지정되는 것이다.

이후부터 C300번지에 넣는 데이터는 PIA의 A포트를 통해서 입출력 된다. 같은 방법으로 B포트도 초기화 시키고 나면 B포트는 C302번지로 지정된다. 즉 C302번지에 넣는 데이터는 PIA의 B포트를 통해서 입출력된다.

이제부터는 애플의 슬롯단자와 PIA 단자와의 연결인데 **그림 6**을 보면서 몇군데만 주의하면 무난할 것이다.

우선 PIA칩 자체를 선택하기 위해 CS0, CS1, $\overline{CS2}$을 이용하는데 PIA가 여러개일 경우 CS을 CPU의 어드레스라인에 연결하여 각각을 선택할 수 있겠으나 본 제작에서는 PIA가 한개뿐이므로 $CS_0 = CS_1 = 5V$에, $\overline{CS_2} = 0V$에 연결만 하면 된다. 다음에 PIA의 E(Enable)단자는 애플의 클럭 $\Phi_0$에 연결하고 $RS_0 = A_0$, $RS_1 = A_1$에 연결하면 된다. 그리고 2개의 IRQ A, B는 OR게이트에 의해 애플의 IRQ에 연결시킨다. 이상의 내용에 주의해서 연결하면 인터페이스 부분에서는 별다른 문제는 없을 것이다.

〔그림 6〕 I/O 핀과 PIA간의 접속도




## 트리거회로

위상제어 회로에서는 교류전원과 제어회로간에 위상이 일치돼야 된다. 위상의 일치를 위해 교류 100V/60Hz의 전원을 트랜스를 이용하여 5V전압으로 만든후 콤퍼레이터IC LM319를 이용하여 트리거시켜 구형파로 만든다. 콤퍼레이터란 비교회로를 말하는 것으로 **그림 7**(a)를 보자.

LM319의 +입력(비반전입력)에 교류를 넣고 −입력(반전입력)에 0전위를 넣으며 −입력의 0전위에 비교하여 출력단에 그림 7(b)와 같은 파형이 나타난다. 이 파형은 상승·하강 검출회로를 통하여 1/120초에 한번씩 리세트 신호를 발생하여 클럭-분주회로와 카운터를 클리어시킨다.

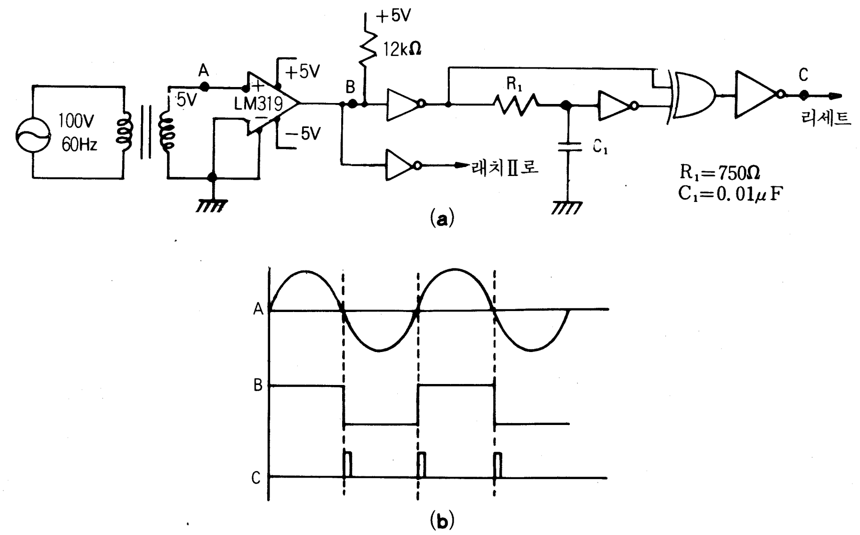
## 카운터회로와 클럭 분주회로

클럭발생기는 7404 TTL IC NOT게이트를 사용하여 수정발진시키고 이 발진파형을 CMOS-IC 4040으로 $1/2^7$ 분주한다. 분주된 주파수는 30.72kHz로 $120 \times 256 = 30.72$(kHz)의 값으로 나타난다. 이는 60Hz전원의 반주기마다 0~256을 카운트 하기 위한 것으로 여기서는 74193 TTL IC를 사용하여 다운카운터로 사용하고 있다. 다운동작을 시키는 이유는 컴퓨터의 데이터값이 클때 전력을 크게 하기 위해서이다.

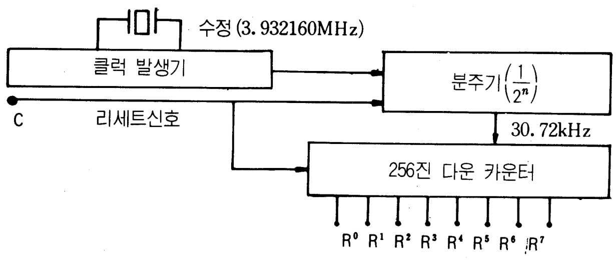
## 비교-래치회로

**그림 9**를 보면 6821PIA의 어드레스 포트에서 74138 디코더를 통해 선택된 조명기구만 인에이블 되어 PIA 데이터 포트의 신호가 래치I을 통과하여 이 값과 카운터의 값이 비교회로에서 비교되어 두값이 같을 때만 래치 II을 인에이블시켜 60Hz의 구형파를 제

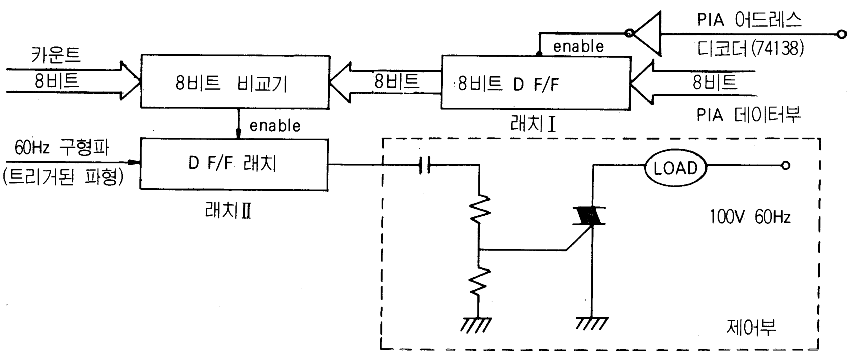
〔그림 7〕 트리거회로와 각 부의 파형




〔그림 8〕 클럭과 카운터




〔그림 9〕 비교래치회로




어부로 출력시킨다.

여기서 래치의 동작을 설명하면 **그림 10**에서와 같이 입력 D에 파형과 같이 넣고 인에이블의 상승에지(Edge) 부분에서 입력 D의 값이 출력 Q에 나타난다. 이 값은 다음 인에이블신호가 발생하기전까지 유지되며 이러한 기능을 하는 것이 래치이다.

## 제어부

그림 9의 점선 부분은 트라이액의 구동부분으로 동작은 **그림** 11의 타이밍표에 나타난다.

컴퓨터로 부터 출력되는 데이터와 비교해서 같으면 인에이블신호가 나와서 래치II를 출력시키고 이 출력은 트라이액을 구동시킨다. 이때 비교시간 만큼의 시간지연이 생겨 전구의 밝기가 달라진다. 예를 들면 데이터가 127 즉 $10000000_{(2)}$이라면 0~127까지 카운터가 수를 세는 동안 시간지연이 생긴후 인에이블신호가 나와 **그림** 11과 같이 반의 전력이 된다. 즉 127까지 세는 동안 1/240초의 지연이 생긴다.

**리스트** 1 프로그램은 가장 기본적인 샘플 프로그램으로 빛이 서서히 밝아졌다 어두워지도록 한 것이다.

프로그램의 작성은 애플 어셈블러인 LISA2.5를 사용하였고 프로그램 옆에 간단한 설명을 적었으니 참고하기 바란다.

이러한 제어원리를 이해하려면 각자가 자기의 용도에 맞게 프로그램을 작성할 수 있을 것이다.

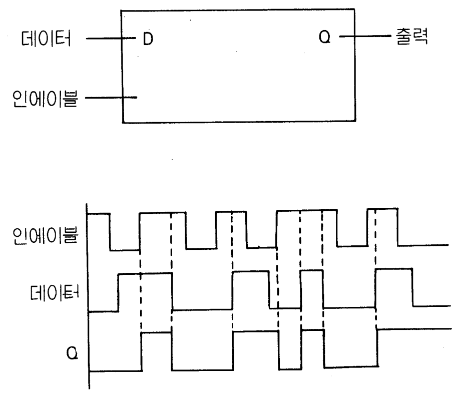
## 맺음말

실제 사용해보니 동작상의 데이터의 값은 20~240사이였다. 이는 트리거회로와 카운터, 래치 등의 IC지연 시간에 의한 것과 트라이액의 도통전압이 수볼트인 관계로 컴퓨터의 데이터 0~255가 정확히 맞지 않았다. 하지만 실제 동작상의 전력은 아주 적게 차이가 나므로 문제가 되지 않는다.

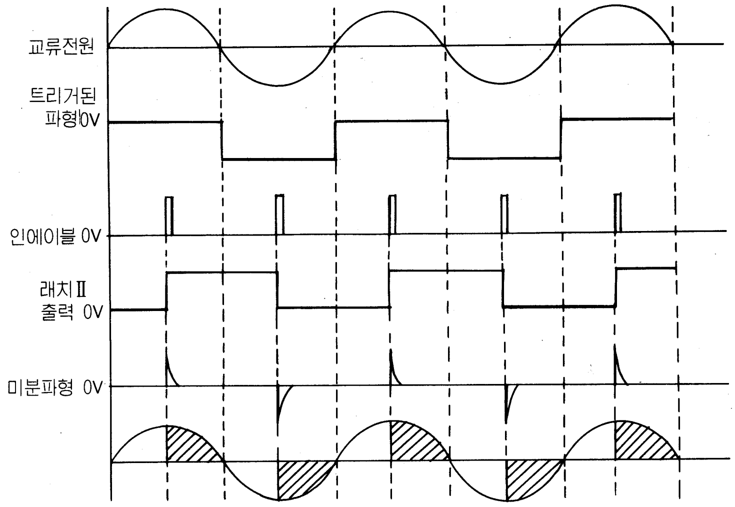
다만 제어회로에서 직접 트라이액을 구동하니 노이즈가 조금씩 발생하였다. 이는 오동작의 원인이 될 수 있으므로 실용상의 회로에서는 포토커플러*로 분리시켜 트라이액에서 발생하는 노이즈가 영향을 받지 않도록 해야 한다.

* **포토커플러 Photo coupler)** : 발광소자와 수광소자를 마주보게 한 간단한 복합소자를 말한다. 입·출력은 전기신호 소자의 내부에서는 빛으로 신호가 전해지게 되어 있다. 기능은 릴레이나 트랜스와 같은 구실을 하지만 대전력을 전달할 수는 없다.

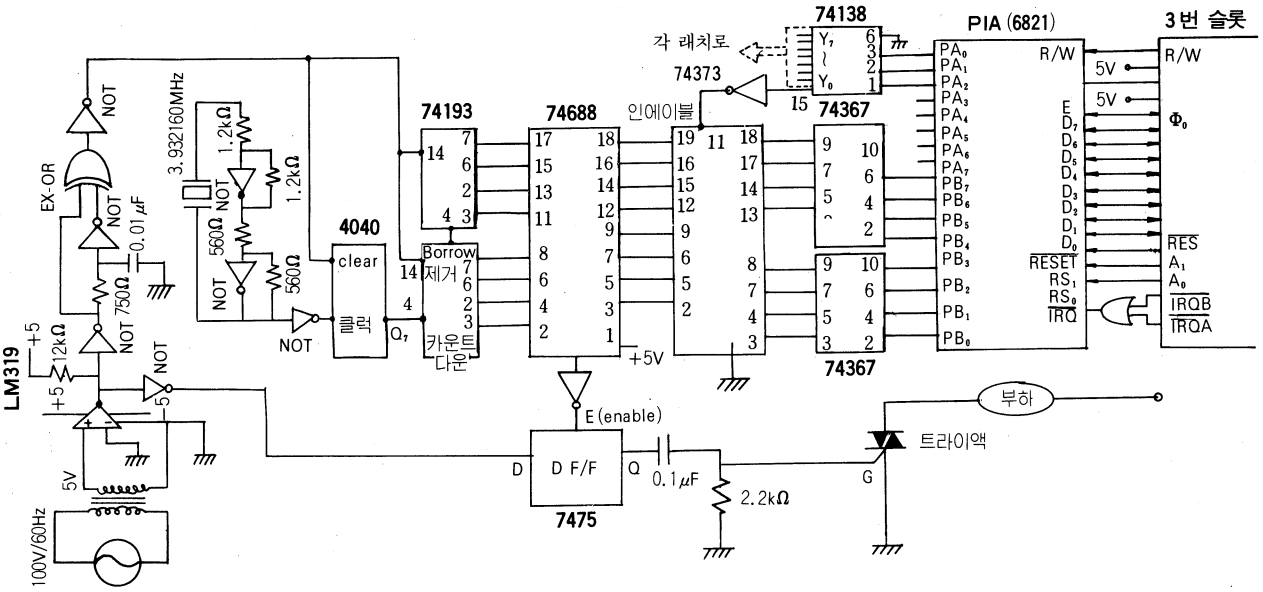
〔그림 10〕 래치의 작용




〔그림 11〕 각 부분의 파형




〔그림 12〕 전체 회로도




〔리스트 1〕 샘플 프로그램

```
0C33                 ORG $C33
0C33                 OBJ $C33
0C33            ** A-port INIT **
0C33 A9 00           LDA #$00
0C35 8D 01 C3        STA $C301
0C38 A9 FF           LDA #$FF
0C3A 8D 00 C3        STA $C300
0C3D A9 04           LDA #$04
0C3F 8D 01 C3        STA $C301
0C42            ** B-port INIT **
0C42 A9 00           LDA #$00
0C44 8D 03 C3        STA $C303
0C47 A9 FF           LDA #$FF
0C49 8D 02 C3        STA $C302
0C4C A9 04           LDA #$04
0C4E 8D 03 C3        STA $C303
0C51            *** INIT END ***
0C51 A9 00           LDA #$00
0C53 8D 00 C3        STA $C300 ;Select Dimmer 00
0C56 A9 00      START LDA #$00
```

```
0C58 8D 02 C3          STA $C302
0C5B A9 70       LOOP  LDA #$70
0C5D 20 A8 FC          JSR $FCA8 ;Delay
0C60 EE 02 C3          INC $C302 ;Increase Data
0C63 AD 02 C3          LDA $C302
0C66 C9 FF             CMP #$FF  ;Compare Data
0C68 F0 03             BEQ FINAL
0C6A 4C 5B 0C          JMP LOOP  ;Loop
0C6D A9 70       FINAL LDA #$70
0C6F 20 A8 FC          JSR $FCA8;Delay
0C72 CE 02 C3          DEC $C302;Decrease Data
0C75 AD 02 C3          LDA $C302
0C78 C9 00             CMP #$00
0C7A F0 DA             BEQ START;Branch To START
0C7C 4C 6D 0C          JMP FINAL;Loop
0C7F                   END
```

# 하드디스크의 셋업

컴퓨터의 주변장치중 빠른 데이터 처리속도와 대용량성이란 특성으로 인해 사용자의 인기를 얻고 있는 하드 디스크는 플로피디스크가 비교적 사용하기 간단한데 비해, 한번의 포맷으로 인해 대량의 정보가 손실되는 등 사용이 까다로와 사용자와 친숙하지 못하다. 만일 독자가 하드디스크를 구입하게 되었을 때나 하드디스크의 용량이 가득차서 정리할 때라든지, 문제가 발생하여 하드디스크를 셋업해야 할 경우, 독자 스스로 할 수 있다면 하드디스크를 더욱 더 친숙하고 편리하게 이용할 수 있을 것이다. 이런 이유로 하드디스크를 셋업시키는 과정을 설명하여 하드디스크를 사용하는데 도움이 되고자 한다.

## 도스 파티션

하드디스크는 하나 또는 그 이상을 가질 수 있는데, 여기서의 설명은 주로 하나의 플로피디스크 드라이브가 준비된 상태에서 하나의 하드디스크를 사용할 경우를 예로 들기로 한다.

하드디스크를 사용할 수 있게 해 주는 OS가 도스인데 이 도스를 하드디스크에 설치하는 과정이 곧 하드디스크의 셋업 과정이다(여기서 사용하는 도스는 버전 3.3이다).

일반적으로 많이 사용되는 하드디스크는 10MB ~ 80MB 정도인데 도스 버전 3.3 이전의 것은 32MB 이상되는 하드디스크를 효율적으로 지원하지 못했다. 그러나 버전 3.3으로 버전업 되면서 하나의 하드디스크에 논리적인 드라이브(C:, D:, E:, F: 등)명을 지정하게 하여 32MB 이상의 하드디스크를 효율적으로 사용할 수 있게 했다.

그러나 32MB 이상의 경우 도스 파티션을 형성하는 과정이 필요한데 이 도스 파티션은 하드디스크상의 파티션이라고 불리는 분할영역중의 하나이다. 하드디스크의 파티션에는 특정 오퍼레이팅 시스템(운영체제)이 존재하는데 이 영역에는 하나 이상의 오퍼레이팅 시스템이 동시에 존재할 수 있으나 실행시킬 때는 하나만 사용할 수 있다. 예를 들어 도스와 IBM의 제닉스가 같은 하드디스크에 설치되어 있어도 동시에 실행시킬 수 없다는 의미이다.

또한 오퍼레이팅 시스템은 하나 이상의 파티션으로 구성할 수 있는데, 그중에 하나는 프라이머리 도스 파티션으로, 이 파티션은 최대 32MB의 크기를 가질 수 있고 하드디스크에서 도스를 로드시켜 드라이브를 사용할 수 있게 하여 준다.

나머지 하나는 익스텐디드 도스 파티션으로 이 파티션은 어떤 크기라도 가질 수 있는데 이것은 다시 논리적 드라이브(Logical Drive)라는 복수의 영역으로 세분화된다. 이 논리적 드라이브는 최대 32MB의 크기를 가질 수 있는데, 만일 디스크가 32MB의 용량을 넘어설 경우 익스텐디드 도스 파티션을 형성하고 다시 논리적 드라이브로 분할한다. 논리적 드라이브는 익스텐디드 도스 파티션의 공간 크기에 따라 드라이브 문자 A에서 Z까지 형성시킬 수 있다.

도스를 시동시키면 각각의 논리적 드라이브에는 드라이브 문자가 지정되고 개별적인 하드디스크처럼 드라이브 문자를 사용하여 그 드라이브에 액세스할 수 있다. 예를 들어 하나의 하드디스크를 가지고 있을 때 프라이머리 도스 파티션과 익스텐디드 도스 파티션을 형성시킨 다음, 프라이머리 도스 파티션에 의해 정의된 영역에 드라이브 문자 C를 지정하고 논리적 드라이브에 정의된 영역에 D를 지정한다.

## FDISK 명령어

이와 같이 하드디스크상에 프라이머리 도스 파티션과 익스텐디드 도스 파티션을 논리적으로 분할해 주는 과정이 필요한데 이 과정은 FDISK 명령어로 할 수 있다. 여기서 주의할 것은 프라이머리 도스 파티션과 익스텐디드 도스 파티션을 사용하기 이전에는 반드시 포맷되어 있어야 한다(포맷을 통해 디렉토리, 화일 할당 테이블 및 시스템 로더를 초기화하여 도스 화일을 받아 들이도록 디스크가 준비되기 때문이다).

FDISK 프로그램은 도스 파티션을 형성하고, 삭제하고, 변경하고, 정보를 표시하는 등 기타 하드디스크의 작업을 실행시키는데 여기서는 도스 파티션 형성 과정만 설명하겠다. FDISK를 시동시키려면 도스 디스켓 #1을 드라이브 A에 넣고 도스 프롬프트 A>가 나타나면

```
fdisk
```

라고 타자하고 리턴키를 누르면 다음의 **화면 1**이 나타난다.

초기화면에서는 옵션이 4개가 나타나는데, 이는 하나의 하드디스크를 가지고 있을 경우이고 하나 이상일 경우는 선택항목 "5. Select Next Fixed Disk Drive"가 하나 더 나타난다. 여기서 도스 파티션을 형성하기 위해 디폴트값 1번을 선택하는 리턴키를 누르면 **화면 2**가 나타난다.

화면 2에서 프라이머리 도스 파티션을 형성시키려면 리턴키를 눌러 디폴트값 1을 선택하면 다음의 **화면 3**을 얻게 된다.

여기서 프라이머리 파티션의 최대 가용공간인 32MB 영역을 모두 사용하지 않고 일부만을 사용할 경우는 N을 선택하고 리턴을 누르면 다음과 같은 **화면 4**가 나타난다.

여기서 "Total disk space"는 전체 디스크상의 실린더수를 나타내는 것이며 "Maximum space available"은 파티션을 할 수 있는 나머지 실린더수를 나타낸다. 가용공간의 일부를 도스에서 사용하고자 하면 실린더수를 타자하고 리턴키를 누른다. 이때는 먼저번의 화면과는 다르게 도스 파티션을 액티브시키지 못하기 때문에 도스 파티션을 형성시킨 후 사용하기 전에 우선 FDISK 옵션 메뉴의 선택항목 2번인 "Active partition"을 변경시켜야 한다. **화면 5**의 경우는 2개의 파티션을 갖는 하드디스크의 파티션상태이다.

액티브 파티션을 선택하려면 파티션의 수를 타자하는데 여기서는 도스 파티션을 액티브해야 하므로 도스 파티션의 수를 타자하고 리턴한다. 이제 이 파티션이 액티브 된 것이다. ESC키를 눌러 FDISK 옵션 메뉴에서 DOS로 되돌아온다.

만일 프라이머리 도스 파티션을 위해 32MB 까지의 최대 가용공간을 사용하고 도스를 하드디스크에서 시동시키려면 리턴키를 눌러 다음 **화면 6**이 나타나게 한다.

도스가 재시작되고 날짜와 시간을 입력할 것을 프롬프트한다. 도스 디스켓 #1이 드라이브 A에 있는 상태에서 날짜와 시간을 입력하면 도스 파티션이 형성되고 도스 프롬프트가 나온다. 이제는 도스를 하드디스크에 설치해야 하는데 이것은 SELECT 명령어를 사용한다.

## SELECT 명령어

SELECT는 키보드 타입과 날짜 및 시간 형식을 도스에게 알리고 도스 파티션을 포맷하고 도스 화일을 하드디스크에 카피한다.

SELECT 절차를 실행동안 도스 파티션이 포맷되므로 하드디스크상에 구 버전의 도스나 기타정보를 가지고 있는 경우에는 SELECT를 사용하면 안되며, 도스 파티션이 10MB보다 크고 도스 버전 3.3으로 포맷된 경우는 하드디스크를 액세스할 때 도스 3.3 이전의 버전을 사용할 수 없다.

그러면 SELECT를 실행시켜 보자. A> 프롬프트에 대해

〔화면 1〕 FDISK 초기화면

```
IBM Personal Computer
Fixed Disk Setup Program Version 3.30
(C)Copyright IBM Corp. 1983,1987

FDISK Options

Current Fixed Disk Drive: 1

Choose one of the following:

    1.  Create DOS Partition
    2.  Change Active Partition
    3.  Dedete DOS Partition
    4.  Display Partition

Enter choice:[1]


Press ESC to return to DOS
```

〔화면 2〕 DOS파티션 선택화면

```
Create DOS Partition

Current Fixed Disk Drive: 1

    1. Create Primary DOS partition
    2. Create Extended DOS partition

Enter choice: [1]

Press ESC to return to FDISK Options
```

〔화면 3〕 DOS파티션 생성화면

```
Create Primary DOS Partition

Current Fixed Disk Drive: 1

Do you wish to use the maximum size
for a DOS partition and make the DOS
partition active (Y/N).............? [Y]

Press ESC to return to FDISK Options
```

**select c: xxx yy**

(여기서 c는 대상 드라이브로 지정하고 xxx는 나라코드이고 yy는 키보드 코드이다)

리턴키를 누르면 다음의 **화면 7**이 나타난다.

여기서 정보가 실려 있으면 N을 타자하고(Y나 리턴키를 타자할 경우 이전에 기록된 정보가 모두 지워져 복구할 수 없게 되기 때문에) SELECT 절차를 시작하려면 리턴키를 누르고 다음 **화면 8**이 나타나게 한다.

Y를 타자하고 리턴하면 포맷팅이 시작되고 화면에 헤드 및 실린더 번호가 나타나고, 이들 번호는 포맷팅 진행에 따라 변한다.

포맷이 완료되면 다음의 **화면 9**가 나타난다.

여기서 Volume label은 하드디스크를 식별하기 위해 사용되는데 만일 Volume label을 입력하려면 최대 11자까지 이름을 타자하고 리턴하며, label이 필요없으면 그냥 리턴한다. 그러면 디스크 현황이 모니터에 나타난다.

여기서 맨 아래의 메시지는 도스화일이 하드디스크에 카피되고 있음을 나타낸다. 도스화일이 카피됨에 따라 화일의 이름이 나타나고 모두 카피되면 A> 프롬프트가 나타난다. 이상의 과정을 거쳐서 하드디스크의 SELECT 절차가 완료되었다.

이제는 하드디스크를 이용해서 DOS를 시동시킬 수 있고, 화일을 구성할 수도 있게 되었다.

만일 사용하고 있는 하드디스크를 다시 셋업시켜야 할 경우 주의해야 할 점은 하드디스크상의 중요한 정보는 항상 백업을 받아 놓아야 한다는 사실이다. 만약 그렇지 않을 경우 순간적인 작은 실수로 모든 데이터를 지워 버리는 불상사가 발생할 수도 있기 때문이다. PC

〔화면 4〕 32MB 전체를 사용하지 않을 경우

```
Create Primary DOS Partition

Current Fixed Disk Drive: 1

No partition defined

Total disk space is  305 cylinders.
Maximum space available for partition
is  305 cylinders.

Enter partition size............: [ 305]

Press ESC to return to FDISK options
```

〔화면 5〕 2개의 파티션을 가진 경우

```
Change Active Partition

Current Fixed Disk Drive: 1

Partition Status  Type    Start End Size
C:  1        A     DOS     000  149  150
    2            EXT DOS  150  304  155

Total disk space is  305  cylinders

Enter the number of the partition you
want to make active...............: [ ]
```

〔화면 6〕 DOS를 하드디스크에서 시동시키려면

```
System will now restart

Insert DOS diskette in dtive A:
Press any key when ready . . .
```

〔화면 7〕 셀렉트절차

```
SELECT is used to install DOS the first
time.  SELECT erase everything on the
specified target and then installs DOS.
Do you want to continue (Y/N)?  Y
```

〔화면 8〕 하드디스크의 포매팅

```
WARNING, ALL DATA ON NON-REMOVABLE DISK
DRIVE c: WILL BE LOST!
Procced with Format  (Y/N)?
```

〔화면 9〕 포맷후의 화면

```
Format complete
System transffered

Volume lable (11 characters, ENTER for none)?
```

# 배치화일을 만들어 보자

PC를 오래 쓰다보면 반복되는 일련의 조작 또는 명령어를 하나하나 입력하는 번거로움을 종종 느낀다. 특히, 프로그램을 개발할 때 컴파일작업과 링크작업은 그 빈도수가 많으며, 또 복잡한 절차와 시간을 요한다는 점에서 쉽게 싫증을 느끼기도 한다.

이때 배치화일(Batch File)을 적절하게 만들어 놓고 활용하면 반복되는 일련의 조작을 컴퓨터가 자동적으로 실행시켜 주기 때문에 상당히 편리하게 된다. 비록 몇가지 제어구조나 명령어를 익히는데 약간 복잡하기는 하나 꾸준히 연습해서 어느 정도 숙달되면 멋있는 DOS 테크닉을 구사할 수 있게 된다.

여기서는 현재 작업중인 디렉토리의 상태를 나타내는 도스프롬프트의 세트방법과 컴파일과정 및 링크작업을 자동적으로 실행시켜 주는 배치화일을 사용자가 직접 세트시키는 방법을 보였다.

## 1. 배치화일에 대하여

모든 배치화일은 반드시 확장자명을 .BAT로 가져야 하며, 특히 전체 화일이름이 AUTOEXEC .BAT로 된 것은 DOS가 부팅된 후에 자동적으로 실행되는 배치화일임을 알아두자. 그외의 경우는 화일명을 입력함으로써 실행이 시작된다(이때 확장자명인 .BAT는 입력하지 않아도 상관없다.).

## 2. 배치화일 작성하기

에디터(Editor)를 사용하거나 표준 입력장치로부터 직접 복사(Copy)명령을 사용하여 배치화일을 작성할 수 있다. 표준 입력장치를 사용할 경우 다음과 같은 형식으로 입력한다.

```
C>COPY CON <화일이름>.BAT
```

그러면 화면은 명령 입력을 위한 대기상태가 되며, 여기에 원하는 명령을 입력한다. 그리고 입력이 끝났으면 F6 (Ctrl-Z)키를 누룬 후 〈리턴〉키를 눌러 빠져 나온다. 그러면 화일이 저장되면서 다음과 같은 메시지가 화면에 나온다.

```
1 File(s) copied
C>_
```

## 3. 도스프롬프트 조작

**그림 1**에 도스프롬프트 세팅의 한 방법을 보였다. 여기서 ECHO는 배치화일내에서 실행되는 명령어를 화면에 표시하느냐 안하느냐를 결정한다. ECHO OFF를 지정하면 그 이후에 지정되는 명령어가 실행될 때 나타나지 않고 그 결과만이 디스플레이된다. 이 ECHO OFF란 명령어 자신도 나타나지 않게 하려면 @ECHO OFF와 같이 앞에 @기호를 붙이면 된다.

또, ECHO 〈메시지〉를 지정하면 ECHO의 ON, OFF에 관계없이 표준 입력장치에 〈메시지〉가 디스플레이된다.

그 다음 CLS에 의해 화면이 지워지고 다음 명령어 PROMPT $P$G가 실행된다. PROMPT란 명령어에는 특수

〈그림 1〉 도스프롬프트 세트

```
C>COPY CON AUTOEXEC.BAT
  ECHO OFF
  CLS
  PROMPT $P$G
  ^Z

C>AUTOEXEC
```

〈그림 2〉 ECHO를 이용한 배치화일

```
C>COPY CON TEST.BAT
  @ECHO OFF
  CLS
  DIR/W
  ECHO  ******************************
  ECHO  *  This Screen Is            *
  ECHO  *    Output Of The BatchFile *
  ECHO  ******************************
  ^Z

C>TEST
```

메타언어(Meta String)를 쓸 수 있는데 그 리스트는 표 1에 나와 있다.

이제 그림 1의 '$P$G'가 디폴트 드라이브의 현 디렉토리와 '>'를 프롬프트로 세팅한 것임을 알 것이다. 이후로 재부트(Ctrl+Alt+Del)시키면 이 AUTOEXEC .BAT가 실행되어 도스프롬프트가 C:\>로 된다. 그리고 어느 서브디렉토리로 들어가든 그 디렉토리의 상태를 항상 프롬프트상에 나타내 준다.

한번 독자들 스스로가 표 1에 있는 문자들을 이용해서 여러가지 도스트롬프트를 설정해 보기 바란다. 이쯤되면 DOS의 위력을 조금은 느꼈을 것이다.

다음은 앞으로 소개할 AUTOCOM .BAT를 위해 %1, %2 같은 배치화일 파라미터에 대해 알아보자. 배치화일 내에서 %1, %2 등을 입력하면 이것은 문자열로 간주하고 차후에 입력되는 문자열과 결합된다. 예를 들어 다음과 같은 내용의 배치화일 TEST .BAT를 생각해 보자.

```
ECHO OFF
ECHO FIRST-%2
ECHO SECOND-%1
```

배치화일 TEST를 'ONE TWO'와 함께 실행시켜보면 다음과 같이 출력된다.

```
ECHO OFF
FIRST-TWO
SECOND-ONE
```

즉, 파라미터 번호순으로 문자열들이 치환됐음을 볼 수 있다. 참고로 %0은 배치화일 자신의 화일이름과 드라이브 이름을 나타냄을 알아두자.

## 4. 컴파일용 배치화일 만들기

웬만한 독자들은 한번쯤 언어를 처리하면서 컴파일작업과 링크작업을 해 보았을 것이다. 특히 이런 작업에서는 많은 절차를 거치게 된다. 예를 들어 어셈블리언어로 작성된 REPORT .ASM 소스화일을 컴파일시키면서 다음을 고려해 보자.

① 일반적으로 어셈블러가 부팅되면서 세번이나 화일이름을 사용자에게 묻는다. 즉 OBJ화일, LST화일, CRF화일이 그것인데 통상 〈리턴〉키를 입력함으로써 공백(Null String)으로 처리해 버린다.

〈예제 1〉 어셈블러가 속해 있는 디렉토리에서 소스화일 REPORT .ASM을 컴파일 시킨 예

```
C:\ASSEMBLE>MASM REPORT
Microsoft (R) Macro Assembler Version 4.00
Copyright (C) Microsoft Corp 1981, 1983, 1984, 1985.

Object filename [REPORT.OBJ]:┐
Source listing  [NUL.LST]:   ├ 모두 〈리턴〉처리한다.
Cross-reference [NUL.CRF]:   ┘

  49632 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

 C:\ASSEMBLE>
```

예제 1에서 일반적인 컴파일과정을 보였다. 화면에서 알 수 있듯이 특정한 지정이 없을 때 어셈블러는 〔 〕속에 디폴트로 정해진 화일 지정을 취한다. 즉, 모두 〈리턴〉처리함으로써 .OBJ화일은 소스화일과 같은 이름(REPORT)으로, LST화일과 .CRF화일은 공백지정으로 처리된다. 컴파일이 완료되면 디렉토리의 목록을 조사해 보라. REPORT .OBJ란 새로운 화일이 생겼음을 볼 수 있다.

더 나아가 이러한 일련의 입력과정을 한 라인에 모아서 처리할 수 있는데 배치화일에서는 이 방법을 응용한다(예제 2).

〈예제 2〉 컴파일시 각 입력과정을 한 라인에 모아 처리한 예

```
C:\ASSEMBLE>MASM REPORT REPORT NUL NUL
Microsoft (R) Macro Assembler Version 4.00
Copyright (C) Microsoft Corp 1981, 1983, 1984, 1985.

  49626 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

 C:\ASSEMBLE>
```

② 컴파일이 완료되면 .OBJ화일이 생성되고, 이 .OBJ화일은 링크작업을 거쳐 .EXE화일을 생성한다. 이제 불필요해진 .OBJ 화일을 삭제한다.

〈표 1〉 메타언어에 이용되는 특수문자

| | |
|---|---|
| $ | $문자 |
| t | 시간 |
| d | 날짜 |
| p | 지정 드라이브의 현재 디렉토리를 나타낸다. |
| v | 도스버전을 나타낸다. |
| n | 지정 드라이브 문자를 나타낸다. |
| g | >문자 |
| l | <문자 |
| b | 문자 |
| q | =문자 |
| ⋮ | |

〈예제 3〉 링크작업 예

```
C:\ASSEMBLE>LINK REPORT

Microsoft (R) 8088 Object Linker Version 3.05
Copyright (C) Microsoft Corp 1983, 1984, 1985.

Run File [REPORT.EXE]:  ┐
List File [NUL.MAP]:    ├ 모두 〈리턴〉 처리한다.
Libraries [.LIB]:       ┘

C:\ASSEMBLE>DEL REPORT.OBJ ····· 이제 불필요해진
                                 .OBJ화일을 삭제한다.

C:\ASSEMBLE>
```

**③ EXE2BIN화일로 .EXE화일을 .COM화일로 변환하고, 또 불필요해진 .EXE화일을 삭제한다.**

결국 최종적으로 REPORT .COM화일만이 남게 되며, 이 화일은 입력 즉시 실행 가능한 상태로 보존된다.

〈예제 4〉 .EXE화일을 .COM화일로 변환하는 예

```
C:\ASSEMBLE>EXE2BIN REPORT REPORT.COM

C:\ASSEMBLE>DEL REPORT.EXE
```

이 세가지 점을 고려하며, 다음의 배치화일을 분석해 보자.

〈그림 3〉 링크용 화일 세트

```
C>COPY CON AUTOLINK
                    ┐
                    ├ 〈리턴〉키를 4번 입력한다.
                    ┘
 ^Z
```

〈그림 4〉 컴파일용 배치화일 세트

```
C>COPY CON AUTOCOM.BAT
  MASM %1 %1 NUL NUL     ··············· ①
  LINK %1@AUTOLINK       ··············· ②
  ERASE %1.BAK           ··············· ③
  ERASE %1.OBJ           ··············· ④
  EXE2BIN %1 %1.COM      ··············· ⑤
  ERASE %1.EXE           ··············· ⑥
  ^Z
```

① 첫번째 %1에 소스화일을, 두번째 %1에 .OBJ화일, 그리고 .LST화일과 .CRF화일은 공백으로 처리
② .EXE화일을 1과정에서 만든 .OBJ화일과 같은 이름으로 하고, 링크화일을 부르고 있다.
〈그림 3〉에서 링크화일이 모두 〈리턴〉 처리되고 있음에 주목
③ 불필요한 백업화일 삭제
④ 컴파일후에 생긴 .OBJ화일 삭제
⑤ .EXE화일을 .COM화일로 전환
⑥ 불필요해진 .EXE화일 삭제
만약 .COM화일로 전환 불가능한 화일일 경우 이 라인은 생략한다.

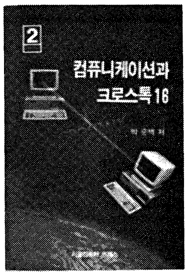
이젠 이 명령 하나로 컴파일과 링크를 단번에 실행할 수 있게 되었다. 이때 소스화일은 확장자명없이 지정해야 하는데 이는 .ASM란 확장자명과 .OBJ 확장자명으로 된 화일을 디폴트로 찾아 할당하는 컴파일러와 링커의 속성 때문이다.

# 신간안내

컴퓨터시리즈 2

## 컴퓨니케이션과 크로스톡16

**박순백 저/서울컴퓨터프레스 발행/**
**정가 5,000원/P.324/국판**



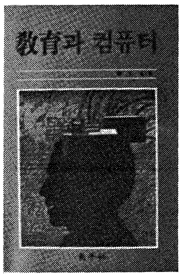

컴퓨니케이션(Compunication)이라는 것은 컴퓨터와 커뮤니케이션이 복합되어 생긴 용어로서 이러한 새로운 용어는 커뮤니케이션 도구로서의 컴퓨터 역할이 커짐에 따라 나타나게 되었다. 본서는 컴퓨니케이션에 관한 필수적이면서도 기본적인 지식에 대하여 서술하고 컴퓨니케이션에서 사용되는 우리말, 영어 용어를 자세히 풀이하여 설명하고 있다.

한편, 제2부에서는 컴퓨니케이션을 행하기 위해 필요한 통신 패키지인 크로스톡16을 소개하고 있다. 크로스톡16은 마이크로스텝사에서 개발해낸 통신 패키지로서 본서는 크로스톡16의 버전 3.6을 채택하여 사용방법을 설명하고 있다. 컴퓨니케이션을 행하고자 하는 사람들에게는 좋은 지침서가 될 것이다.

---

## 교육과 컴퓨터

**유완영 저/교학사 발행/**
**정가 4,500원/P.306/국판**




현대 사회에서 컴퓨터의 역할이 계속 커짐에 따라 학생들에게 컴퓨터를 가르쳐야 한다는 주장이 고조되고 있고, 사회전반에 걸쳐 관심 또한 크다.

그리하여 본서는 교육과 컴퓨터와의 관계를 설정하고, 컴퓨터 교육의 목표와 내용 및 자료의 제시를 목적으로 한다.

제1장~4장은 교육과 컴퓨터와의 관계에서 컴퓨터가 교육을 위하여 어떻게 활용될 수 있느냐는 측면에서 전개하고, 제5장에서 10장까지는 컴퓨터 교육의 목적이 컴퓨터의 기본구성과 원리를 이해시키고, 이를 사용하여 효과적으로 문제를 해결하여 사용자로서의 올바른 태도와 가치관을 형성하는데에 있다고 보고 컴퓨터의 개요, 사용법, 응용 등을 서술하고 또한 컴퓨터와 사회문제에 관해서도 언급하고 있다.

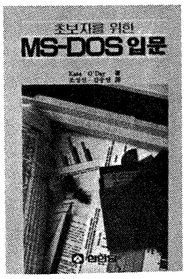
컴퓨터를 가르치는 교사나 컴퓨터를 배우고자 하는 학생들에게 좋은 지침서가 될 것이라고 생각한다.

---

초보자를 위한

## MS-DOS 입문

**조성선·김승연 역/성안당 발행/**
**정가 5,000원/P.378/국판**




MS-DOS는 마이크로소프트사가 개발한 OS로서 마이크로컴퓨터에서 유용하게 사용되고 있다.

본서는 MS-DOS를 이해하기 위해서 컴퓨터가 무엇이며, 컴퓨터는 무엇때문에 오퍼레이팅 시스템을 필요로 하는지에 대한 설명에서부터, 시스템을 작동시키고, 디스켓을 다루며, 시스템에 데이터를 어떻게 입출력시키는가 하는 방법 등을 컴퓨터에 대한 기본 지식이 없는 사람들도 흥미를 갖고 이해할 수 있도록 그림을 곁들여 단계적으로 설명하고 있다.

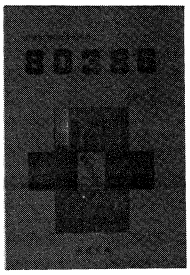
또한 하드디스크 사용자를 위해서 백업과 리스토어 명령 등을 간결하고 명확하게 설명하고 있다.

---

32비트 마이크로컴퓨터

## 80386

**편집부 편/대림출판사 발행/**
**정가 4,500원/P.256/국판**



마이크로컴퓨터의 기술진보에 수반하여 마이크로프로세서는 산업의 여러 분야에서 사용되고 있으며, 그 응용분야도 많아지고 있다.

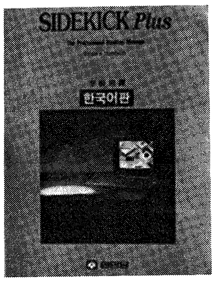
따라서 본서는 급속히 발전을 하는 오늘날의 기술혁신하에서 32비트의 주류가 되는 인텔의 80386에 초점을 두어 32비트 마이크로프로세서라는 것은 어떤 것인가를 설명하고 있다.

한편, 이 책은 8086의 예비지식을 갖고 지금부터 80386의 학습을 시작하는 사람들을 위한 안내서로서 80386의 새로운 개념을 중심으로 그림을 보고 잘 이해할 수 있도록 기술하였다.

---

## SIDEKICK Plus

**이상민 역/성안당 발행/**
**정가 6,000원/P.298/4·6배판**




본서는 볼랜드사 제품인 사이드킥 플러스 메뉴얼을 편역한 것으로서 모두 3부로 구성되어 있다.

제1부는 사이드킥 플러스에의 입문, 제2부는 사이드킥 플러스간의 각 프로그램에 대한 상세한 참조가이드이며, 제3부는 사이드킥 플러스를 커스터마이징하는데 필요한 내용이 담겨 있다.

기존의 사이드킥보다 강력하고 다양한 기능을 가지고 있는 사이드킥 플러스를 사용하고자 하는 이들에게는 좋은 참고서가 되리라 본다.

---

컴퓨터시리즈 1

## 컴퓨터 학습

**임경선 저/도서산업사 발행/**
**정가 3,000원/P.219/국판**

정부에서 1983년을 정보산업 육성의



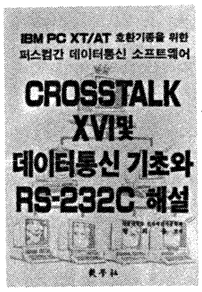

해로 설정한 이래 국내에서도 본격적인 컴퓨터의 열풍이 불기 시작하였다. 따라서 컴퓨터의 활용범위가 넓어지고 특수계층만의 전유물이였던 컴퓨터에서 대중화된 컴퓨터로 개념조차 바뀌어 가고 있는 실정이다.

이에 본서에서는 컴퓨터를 처음 배우고자 하는 사람들이 MSX 전기종의 충분한 활용을 위하여 단계적인 예제와 보충설명을 곁들여 기초에서부터 간단하면서도 이해하기 쉽게 서술하였다.

---

## CROSSTALK XVI 및 데이터통신 기초와 RS-232C 해설

**황희융 저/교학사 발행/**
**정가 8,000원/P.515/4·6배판**




오늘날, 16비트 PC 사용자들이 늘어남에 따라 PC 자체의 기능뿐만 아니라 PC 사용자들간에 서로 정보교환을 요구하게 되었다. 예를 들어, 원거리에 있는 두 컴퓨터 시스템간의 데이터 전송이나, 시스템 구축을 통한 주변기기의 공유 등을 들 수 있다.

이와같이 컴퓨터를 통한 정보교환에 대한 사용자들의 욕구를 소프트웨어적으로 해결, 충족시키고자 만든 것이 통신 패키지이다.

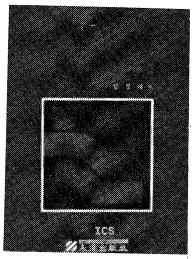
본서에서는 여러 통신 패키지중 사용이 간편하고, 통신상황을 화면으로 쉽게 알 수 있게 해주는 Microstuff사의 CROSSTALK-16을 소개한다.

제1편에서는 CROSSTALK-16의 사용법을 소개하고, 제2편에서는 데이터통신의 기초로서 RS-232C에 관한 설명과 직렬전송제어에 관하여 그림을 곁들여 알기 쉽게 설명하고 있다.

---

프로그래머를 위한
## dBASE III PLUS(Level 2)

**김경태 저/영진출판사 발행/**
**정가 6,000원/P.248/4·6배판**



본서는 dBASU III PLUS Level 1에 이어 출판된 것으로 dBASE III PLUS의 명령과 함수를 프로그래밍 실무 측면에서 활용할 수 있도록 전개하였다.

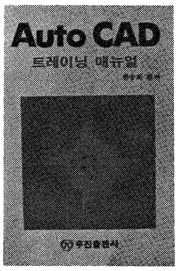
특히 책에 실린 60여개의 단위 프로그림 예제는 dBASE의 명령 기능을 프로그래밍 측면에서 이해하도록 하였고, 직접 어플리케이션을 작성할 시에 참고할 수 있도록 하였다.

또한 dBASE III PLUS의 보완적 방편으로 Ashton-Tate사에서 발표한 dBASE Tools for C의 그래픽과 프로그래머 라이브러리에 대한 운용설명을 덧붙이고 있다.

---

## Auto CAD
트레이닝 메뉴얼

**원송희 편저/우진출판사 발행/**
**정가 5,000원/P.249/국판**




인간에게 있어서 이미지나 도형은 문자에 뒤지지 않는 중요한 메시지의 전달수단으로서 사용되고 있다.

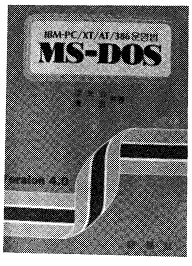
본서는 Auto CAD를 사용하려는 초심자를 위한 안내서로서 제1부는 CAD의 특징과 환경 설정 등에 관하여 적고, 제2부에서는 Auto CAD를 완전히 소화시키기 위한 기본조작을 설명하고 있다.

책의 내용에 따라 직접 조작해 봄으로써 쉽게 Auto CAD에 접근할 수 있으며, 이를 이용한 다양한 응용도 가능할 것이다.

---

IBM-PC/XT/AT/386 운영법
## MS-DOS

**이광치·황 철 공저/연학사 발행/**
**정가 6,500원/P.365/4·6배판**




컴퓨터를 사용하는데 있어서 그 컴퓨터에 맞는 운영체제(OS)는 반드시 필요하게 되는데 IBM-PC/XT나 IBM-PC/AT에는 MS-DOS라고 하는 운영체제가 사용되고 있다.

본서는 MS-DOS 버전 4.0 메뉴얼로서 IBM-PC/XT, IBM-PC/AT를 사용하고자 하는 사람들을 위하여 서술하였고, MS-DOS외에 TURBO OS와 Quick DOS, CP/M 86, 한글 사용법 등을 추가 설명하고 있다.

또한 하드 디스크 드라이브 및 라인 프린터 사용방법에 대해서도 설명하고 아울러 많은 예문을 삽입하여 독자들의 이해를 돕고 있다.

특히 특징적인 것은 직접 XT와 AT를 사용하여 예문을 작성하였다.

---

IBM-PC
## RUN/C 예제와 연습

**박영근 저/대림출판사 발행/**
**정가 5,000원/P.265/4·6배판**




최근에 C 인터프리터인 RUN/C가 발표되었다. 이 RUN/C는 C의 언어

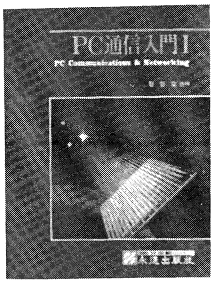
수단을 거의 만족시키고, 더구나 컴파일러같이 귀찮은 작업없이 곧 바로 실행결과를 얻을 수 있어 초보자들이 사용하기에는 매우 편리하다.

본서에서는 C언어의 문법을 연구하는 것보다는 C언어로 프로그램을 작성하기 위한 조작 및 연습에 중점을 두고, 제1장~3장은 각각 RUN/C의 조작법, C언어와 베이직언어와의 비교, C의 명령이란? 순으로 구성되었고, 이 책에서 가장 중점을 둔 제4장은 많은 예제와 문제를 수록하여 프로그래밍 테크닉을 익힐 수 있도록 하였으며, 그리고 마지막장에서는 인터프리터 방식인 RUN/C에서 컴파일러 방식인 Lattice C로 변화되는 것을 중심으로 차이점과 변경점을 설명하고 있다.

한편 본서는 각 장이 전개되기전 간단한 설명을 곁들여 그장의 내용을 한눈에 알아 볼 수 있도록 했다.

## PC통신입문(I)

**정영일 편저/영진출판사 발행/**
**정가 5,000원/P.323/4·6배판**




오늘날 퍼스널컴퓨터가 널리 보급되고, 각종의 우수한 통신 소프트웨어가 개발됨에 따라 데이터통신은 전자 게시판이나 전자사서함 등의 퍼스널 통신시대로 접어들고 있다. PC 유저나 네트워크 매니저는 각종의 우수한 통신 소프트웨어를 이용, 자유롭게 통신을 하지만 통신 프로토콜의 표준화, 고속도 모뎀 기술의 개발, 전자회선 전송품질의 전면적인 개량 등 해결해야 할 문제점들이 많은 것 같다.

본서는 데이터통신의 소개, 데이터통신의 기초, 통신코드 및 제어문자 등에 대하여 설명하고 그것에 관련된 모든 하드웨어와 소프트웨어를 소개하고, 프로토콜 및 네트워크 아키텍처 등에 관해서도 언급하고 있다.

또한 부록에는 데이컴 통신서비스를 제공받으려는 사용자를 위한 가이드가 적혀 있다.

## C언어 500제

**이준구 편저/크라운출판사 발행/**
**정가 5,000원/P.285/4·6배판**



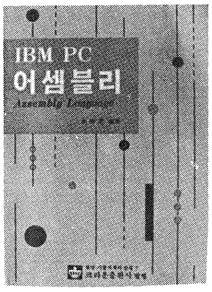

오늘날 퍼스널컴퓨터의 사용자가 많아지므로서 C언어의 버전 또한 다양하게 발전하고 있다.

C언어는 포트란, 코볼, 베이직 등의 언어가 해결해 주지 못하는 역할을 완벽하게 처리해 주고 있으므로 C언어를 사용하는 사람들이 많아지고 있다. 국내에서는 마이크로소프트사의 Micro-Soft C, 볼랜드사의 Turbo-C가 주류를 이루고 있으며, Lattice C, Quick-C 등도 많이 사용되고 있다.

이들에 대한 명령어와 내장함수의 차이는 조금 있으나 본서는 공통적인 부분을 작성하였으며, 500개의 예제로 C언어의 입문에서 실습에 이르기까지 언어에 관한 제반내용을 예를 들어 프로그램으로 설명하여 쉽게 이해할 수 있게 하였다.

## IBM PC 어셈블리

**이준구 편저/크라운출판사 발행/**
**정가 6,000원/P.334/4·6배판**




어셈블리언어는 컴퓨터언어의 구조를 가지면서도 여러 종류의 컴퓨터는 물론 각종 프로그램을 위한 툴로 사용되어 왔다. 그리고 컴퓨터나 제어기기에 사용되는 CPU의 처리체계에 따라 매우 상이한 점을 많이 소유하고 있다.

본서에서는 IBM사의 퍼스널컴퓨터 기종인 PC/XT/AT/PS 2 등에서 공통적으로 사용되는 부분을 다루었고, 또한 80286과 8088의 차이점에 대해서도 서술하였다.

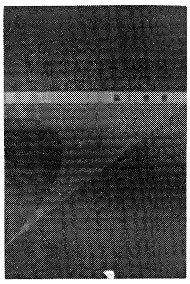
내용을 보면 어셈블리어로 프로그램을 작성시 디버깅에 필요한 DEBEG에 대한 것과 어셈블리와 같이 사용되는 용법인 .COM 화일과 EXE 화일의 차이점을 설명하였다.

주변장치인 키보드 처리, 화면처리, 디스크 처리 등과 프로그램 작성시 꼭 필요로 하는 화일처리에 관한 것, 좀 더 전문적인 프로그램 작성을 위한 Pseudo OP 사용법에 대해서도 소개하였다.

또한 부록에는 어셈블리어로 프로그램을 작성할 때 꼭 알아야 할 BIOS, DOS 참고부분, 디바이스 드라이버, IOCTL 디스켓에 대해서도 소개하고 있다.

## 기사1, 2급 전자계산기조직 응용

**나인화 저/청문각 발행/**
**정가 8,000원/P.797/국판**



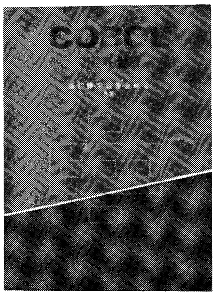
정보산업의 발전에 따라 자료처리분야가 크게는 기업경영에서부터 일상생활에 이르기까지 컴퓨터에 대한 관심이 날로 광범위하게 늘어나고 있다.

이 점을 감안하여 본서는 전자계산기 조직응용 기사시험 1, 2급에 대비하여 유사과목인 전자계산기 프로그래밍 일반, 전자계산기 일반, 시스템 프로그래밍 일반, 자료구조 일반, 어셈블리 프로그래밍 일반, 논리회로 일반, 마이크로 전자계산기, 데이터통신에 대한 철저한 문제분석과 상세한 풀이를 하였고, 그리하여 어떤 유형의 문제를 접하더라도 응용력을 배양할 수 있도록 하였다.

또한 부록에는 시험요강에 대한 자세한 안내가 되어 있다.

## COBOL (이론과 실제)

**나인화 외 2인 공저/청문각 발행/**
**정가 6,500원/P.409/4·6배판**



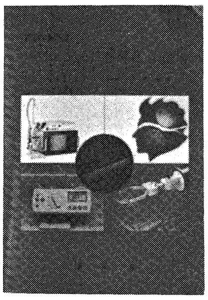

COBOL은 데이터가 많은 자료를 처리하기에 적합한 컴퓨터언어로 사무계산 및 상업계산용에 사용할 수 있는 프로그램 언어이다. 또한 기종간의 별 차이없이 공통으로 사용할 수 있어 전 세계적으로 가장 많이 사용하고 있는 컴퓨터언어이다.

본서는 COBOL의 기본적인 이론과 체계를 곧 바로 실무에 응용할 수 있도록 구성되어 있으며, COBOL문법을 자세하게 설명하여 체계를 이해하도록 각 장마다 본문의 문법과 관련된 많은 프로그램 예제를 실어 응용력을 배양하도록 하였고, COBOL 언어로 업무처리를 함에 있어 업무 내용의 변동에 따라 프로그램 보수가 쉽게 이루어지도록 중대한 문제가 선결되어야 할 경우 구조적 COBOL이 널리 사용되도록 그 내용도 편성되어 있다.

Z80 중심의
## 마이크로프로세서와 그 응용

**보달복 저/청문각 발행/**
**정가 7,300원/P.458/4·6배판**

1971년 인텔사에서 최초의 마이크로프로세서 Intel 4004가 개발된 이래 마이크로프로세서는 그 자체의 개발과 그

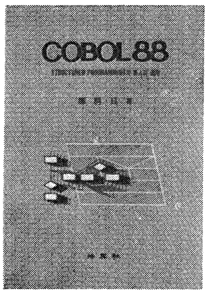


응용의 개발면에서 경이적인 발전이 거듭되어 컴퓨터 관련 기술자 뿐만 아니라 일반 전자, 전기, 기계 기술자도 이에 대한 지식없이는 안되게 되었으며, 이에 부응하여 이 책은 Z80 중심으로 기초에서부터 응용에까지 이해하기 쉽게 쓰여졌으며, 초보자라도 곧 바로 마이크로프로세서를 이용하여 간단한 제어실험을 할 수 있도록 하였다.

제1장은 마이크로프로세서의 개요, 2장 마이크로컴퓨터의 기본 구성, 3장 마이크로컴퓨터의 기본동작, 4장 소프트웨어, 5장 기본 프로그램, 6장 컴퓨터의 하드웨어, 7장 마이크로컴퓨터에 의한 제어 등으로 구성되어 있다.

## COBOL 88

**정병일 저/법영사 발행/**
**정가 8,500원/P.502/4·6배판**




PROGRAM을 설계하고 개발하여 똑같은 처리결과를 이끌어 내는데 있어 그 논리의 개발방법은 사람마다 다를 수 있을 것이다. 그러나 환경을 바탕으로 하는 정보처리는 환경이 부단히 변화함에 따라 거기에 맞도록 프로그램을 유지·보수해 나갈 수 밖에 없기 때문에 개발 당사자외의 실무자에게는 이해하기 어려운 프로그램이 되어 유지·보수에 많은 시간과 비용이 소요된다.

여기에 표준화된 프로그램 개발기법의 필요성이 대두된다. 이에 맞추어 본서는 다음과 같이 구성되었다.

제1편에서는 COBOL 전반에 대한 기초지식을 다루었고, 제2편에서는 구체적인 논리유형 가운데 전산업무 현업 실무자들에게 꼭 필요하다고 생각되는 READ & WRITE, RECORD STACKING, USTING, 통계자료, SORTING, EDITING, FILE MERGING, RAMDOM처리에 대하여 프로그램 구조화의 방향을 제시하고 실제 업무로그 적용사례와 응용예제를 다루었다. 제3편은 부록편으로 '85 ANS, '81 CODASYL, '85 JIS COBOL 예약어, COBOL 명령어를 요약 정리하여 수록하였다.

## 터보 파스칼 4.0

**전용진 편저/크라운출판사 발행/**
**정가 8,000원/P.507/4·6배판**




Turbo Pascal은 IBM PC 및 호환기종을 사용하는 이용자들의 다양한 욕구를 충족시키고자 개발된 언어로 모든 유형의 응용프로그램을 개발하는데 사용될 수 있는 구조적인 고급언어이다.

본서는 미국 볼랜드사에서 발간한 Turbo Pascal 4.0의 내용을 편역한 것으로서 터보 파스칼 프로그램을 작성하고, 컴파일하고, 세이브하는 순서에 맞추어 구성이 전개되어 있다.

또한 버전 4.0의 새로운 특성과 사용방법에 대해서 설명하고 버전 3.0이하에서 작성된 프로그램이 버전 4.0에서 실행되도록 프로그램을 수정하는 방법에 대해서도 설명되어 있다.

이 책은 세부분으로 구성되는데 제1부는 이용자 가이드, 제2부는 프로시듀어 및 함수 참고, 제3부는 부록으로 구성되어 있다.

## AUTO CAD 실무

**송희선 편저/크라운출판사 발행/**
**정가 5,500원/P.315/4·6배판**

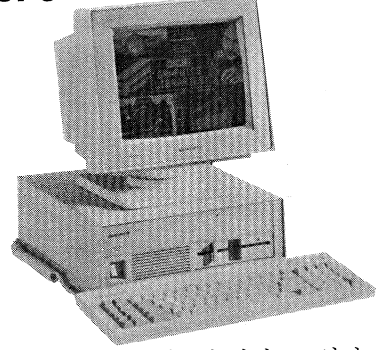



본서는 일반적인 전산기 이용설계 및 제도 프로그램인 AUTO CAD를 써서 기본제도를 하고 인쇄할 수 있는 방법 및 그 내용을 보여주고 있다.

이것은 사용자나 관리자처럼 전산기 이용 설계 및 제도를 쓰기 시작하는 사람을 위한 안내서가 되며, 특히 이 책은 누구나 읽어보기만 해도 쉽게 배울 수 있도록 되어 있다.

# 신제품

## 60만원대 교육용 컴퓨터
## CPC-4000L

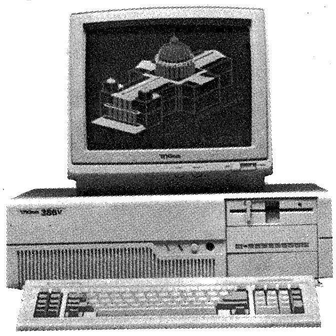


이 시스템은 별도의 칼라보드없이 시스템 내부의 스위치변경으로 칼라모니터를 연결할 수 있으며 640kB까지의 메모리 확장기능과 한글DOS와 KS 표준한글을 내장하고 있다. 그리고 CPU의 속도를 4.77/8/10MHz로 조정이 가능하며 90V에서 250V까지 자유로이 전원을 조절할 수 있으며 제품구입시 자사에서 개발된 워드프로세서와 MUG(Menu Generator)프로그램을 비롯하여 기타 유틸리티 등을 무상으로 제공하고 있다. 제품의 규격은 다음과 같다.

- CPU : 8088-1
- RAM : 256kB(메인보드상에서 640kB 확장가능)
- 보조기억장치 : FDD : 500kB(5.25″)×1, HDD : 20MB(5.25″)(선택)
- 슬롯갯수 : 4
- 시리얼포트 : (선택)
- 패러럴포트 : 1개
- 한글 : KS(국가표준 완성형), 조합형, 대우 DH한글

**제조원 및 판매원 : 대우전자**
**전　화 : (02)754-5637**

## 최첨단 32비트 컴퓨터
## 트라이젬 386V



트라이젬 386V는 최근 개발된 최상위 32비트 PC로 20MHz의 고속처리, 폭넓은 시스템 확장성, 고도의 멀티유저기능 등을 이용하여 종합 OA업무, 과학기술분야, 엔지니어링 등 특수분야의 업무에서도 뛰어난 기능을 발휘할 수 있다.

주요 규격은 다음과 같다.

- CPU : i80386 32비트(20MHz)
- Co-processor : i80387(선택)
- 주기억장치 : RAM 2MB
- 보조기억장치 : FDD5¼″(1.2MB) 1~2개, HDD 40MB, 80MB, 160MB
- 백업장치 : 60MB(XENIX, DOS겸용, 선택)
- 키보드 : 한/영 101키
- 슬롯갯수 : 8개
- 치수(mm) : 538(W)×426(D)×150(H)×150(H)

**제조원 및 판매원 : 삼보컴퓨터**
**전　화 : (02)390-2114**

## CAD/CAM/CAE시스템
## 메가 TOWER/386



TOWER/386은 ACRTC그래픽스 프로세서와 19인치, 20인치의 칼라모니터로 고해상도의 그래픽디스플레이를 제공, CAD/CAM/CAE분야 및 워크스테이션용으로 적합하다.

인텔의 80386 마이크로프로세서를 채택하고 있으며 특히 Auto CAD나 Versa CAD 디자인 등 CAD패키지를 이용하는데 훌륭한 기능을 발휘한다.

제품의 주요 규격을 보면 다음과 같다.

- CPU : Intel80386(16MHz, 20MHz)
- Co-Processor : Intel80387(선택)
- RAM : 2MB(80ns)
- 보조기억장치 : FDD : 1.2MB, HDD : 40MB
- 확장슬롯 : 8개
- 19인치, 20인치 칼라모니터(1280×1024)
- 4096칼라 가능

**제조원 및 판매원 : 메가컴퓨터**
**전　화 : (02)887-8984**

## FA용에 적합한 32비트 컴퓨터
# Bear 386A




Bear 386A는 26MHz의 처리속도와 최고 16MB까지 확장가능한 메모리, 그리고 80387, 또는 80287 연산보조프로세서 추가기능을 갖고 있어서 CAD/CAM/CAE, FA 등 전문분야에서 쓰일 수 있는 32비트 PC이다.

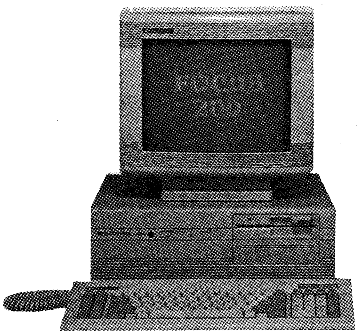
특히, XENIX System V-80386 버전하에서, 멀티-태스킹업무나 많은 입출력 조작이 요구될 때에도 안정된 기능을 유지할 수 있다.

주요 규격을 보면 다음과 같다.

- CPU : 80386
- Co-Processor : 80387, 80287(선택)
- ROM : 256kB(표준)
- RAM : 2MB(표준) (시스템 보드상에서 최고 16MB까지 확장가능)
- 속도 : 16~24MHz(스위치 선택)
- 확장슬롯 갯수 : 124핀×2, 98핀×4, 68핀×2
- 보조기억장치 : FDD : 1.2MB×1
  HDD : 40MB×1
- 콘트롤러 보드 : 2HDD/2FDD콘트롤러 카드
- MGP카드 : 허큘레스+한글카드
- 시리얼/패러럴 포트 : 2 시리얼 & 1 패러럴 포트
- 전원 : 220W, 110V/220V(스위치 선택)
- 모니터 : 14″ 모노크롬
- 키보드 : 101키(12함수키 포함)
- OS : MS-DOS3.30, UCSD-P, UNIX, XENIX 등
- 사용언어 : 포트란, 시, 어셈블러, 파스칼, 로고, PL/1, 등
- 추가(선택) : 칼라그래픽 카드(EGA, PEGA, VEGA, SEGA 등)
  칼라모니터
  마우스 어뎁터
  멀티-펑션카드
  멀티 I/O카드 등

**제조원 및 판매원** : (주)홍익전자
**전 화** : (02)335-0611

## 고속 업무용 컴퓨터
# FOCUS-200




이 제품은 일진전자가 공급한 옵니네트 LAN의 워크스테이션으로 설계된 시스템으로 XENIX를 OS로 사용하면 16명까지 동시에 사용할 수 있다. 또한 한글, 한자는 국가표준코드(KS행정망용 한글한자 카드 내장)을 채택하여 2바이트 완성형으로 처리된다.

주요 규격은 다음과 같다.

- CPU : 인텔80286
- Co-Processor : 인텔 80287
  (12.5MHz)
- 메모리 : ROM : 64kB
  RAM : 1MB(최고 16MB)
- 비디오 보드 : 국가표준코드(K.S 코드), 한글 : 2350자, 한자 : 4888자, 사용자 정의문자 : 188자
- 모니터 : 12″, 14″
- 보조기억장치 : FDD : 1.2MB(5¼ 인치), HDD : 20, 30, 40, 74, 126MB
- 백업장치 : 10, 20, 30, 60MB(선택)
- 슬롯갯수 : 8개
- 시리얼 포트 : 2 RS-232C
- 패러럴 포트 : 1
- 키보드 : IBM PC/AT 호환
- STREAMING 테이프 드라이브 : 1CTD(40MB)

**제조원 및 판매원** : 일진전자(주)
**전 화** : (02)888-6813

## 한글 사용이 가능한 랩탑컴퓨터
# WALKOM



이 제품은 주기억용량이 1MB, 속도가 12MHz의 성능을 갖추고 있으며 80286 마이크로프로세서를 탑재한 AT 호환기종이다.

전원은 90에서 200V까지 자동으로 절환되며 1.44MB의 3.25″와 1.2MB의 5.25″ FDD도 내장되어 있다.

또한 표준슬롯으로 풀사이즈 1개와 하프사이즈 1개 등 2개를 사용할 수 있고 EMS(메모리 확장)기능도 지원된다.

특히 EGA, CGA, 허큘레스 호환가능한 장점도 있다.

또한 동제품은 대만의 Modern Computer Corp와 뉴텍코리아가 기술합작으로 대만에서 생산된 LCD랩탑컴퓨터이다.

**제조원** : Modern Computer Corp.
**판매원** : (주)뉴텍코리아
**전 화** : (02)585-3891

## X-Y 플로터
# EDG-2400

이 X-Y플로터는 소프트웨어를 다양하게 공급하고 저가격화 하여 누구나 쉽게 구입하여 사용할 수 있도록 한 제품이다.

특히 이 제품은 미 HP사의 HP-GL

과 호환성을 가짐으로 Auto CAD, Versa CAD, LOTUS 및 기타 각종 소프트웨어에 사용할 수 있을 뿐만 아니라 한글까지도 플로트(Plot) 할 수 있는 고해상도(High Resolution) 플로터이다.

사용하는 펜은 8가지로 수성펜과 유성펜 등 다양하게 사용할 수 있으며, 사용용지는 420㎜×297㎜이내의 것으로 종이나 OHP 필름 등을 사용할 수 있다.

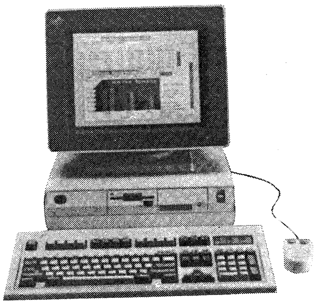
또한 IBM PC XT/AT, AppleII 등을 지원할 수 있으며 시리얼(RS-232C) 및 패러렐(centronics) 인터페이스가 가능하다.

EDG-2400 자체내에는 2.5k바이트의 버퍼를 내장하고 있으며, 별도의 버퍼를 사용할 경우 컴퓨터가 없어도 언제든지 재플로트(Replot) 할 수 있다.

컴퓨터와의 연결은 어느 기종과도 가능하며, 저가격 보급형으로서 CAD시 특히 편리하다.

**제조원 및 판매원 : (주)ED엔지니어링**
**전 화 : (02)582－1981～5**

## 그래픽 기능이 강화된 모델 30/286



이 제품은 강력한 처리능력의 10MHz 80286 프로세서를 표준으로 장착, 기존의 30모델에 비해 약 2배의 빠른 속도를 제공하고, 그래픽 기능에는 과거 상위기종모델에서만 제공되던 향상된 VGA(Video Graphic Array)기능을 제공한다. 또한 기존 PS/2 30모델 용량의 약 25배에 상당하는 16MB까지의 메모리확장성과 상위모델에서 사용되던 3.5인치 고밀도(1.44MB) 디스크장치를 표준 장착하고 있다.

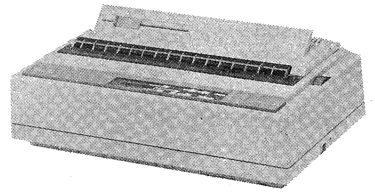
특히, 이 모델과 최근에 IBM에서 발표한 DOS 4.0 등의 오퍼레이팅 시스템과의 조화는 각종 업무에 있어 사용자들에게 더욱 저렴한 가격, 강력한 성능, 손쉬운 사용 등의 이점을 제공한다.

그외 주요 규격은 다음과 같다.

- CPU : 80286(10MHz)
- Co-Processor : 80287(선택)
- RAM : 0.5MB～4MB(120ns), 최고 16MB
- 운영체제 : DOS 3.30 또는 4.0, OS/2
- FDD : 1.44MB(3.5인치)
- HDD : 20MB(80ms)
- 키보드 : 101키
- 크기 : 102㎜×406㎜×397㎜

**제조원 : IBM**
**판매원 : 한국IBM**
**전 화 : (02)781－6941**

## 24핀 도트매트릭스 프린터 DP-2600



한글, 한자 및 풀그래픽이 가능한 24핀 프린터 DP-2600은 유선형의 방음커버를 사용, 소음주파수를 54dB 이하로 저하시킨 제품으로 엽서크기부터 전산용지까지 다양한 크기의 용지를 사용할 수 있고, 다기능의 콘트롤 패널이 있어 셋업모드기능 조작이 가능하며 이물질의 유입을 방지하여 헤드를 보호할 수 있다. 또한 이 제품은 PRO-시리즈 등 IBM PC/XT/AT호환기종, IBM 5550/5540, FA-COM DF-9450 II, BURROUGHS TDI와 접속이 가능하며 IBM 2바이트 한글/한자 코드와 국가행정전산망 KS한글/한자 코드 및 중·대형의 N바이트 코드를 지원할 수 있다.

주요 제품 규격을 보면 다음과 같다.

- 프린트방식 : 24핀 도트매트릭스
- 프린트방향 : 양방향(최단거리 이동)
- 인쇄속도 : 영문/숫자-330CPS(HIGH SPEED DRAFT), 250CPS(DRAFT), 82CPS(LQ)·한글/한자-70CPS(LQ), 140CPS(DRAFT)
- 문자구성 : 영문/숫자—12×8
  한글/한자—24×24
  반각/문자—24×12
- 인자기능 : 밑줄, 굵은문자, 역상, 상하첨자, 음영, 세로 2배확대, 가로세로 2배확대, 다양한 확대기능(1.5배, 2배, 3배, 4배, 6배, 8배, 16배)
- 복사매수 : 원본포함 5매
- 리본 : 롱 카트리지 규격(수명은 400만자)
- 인터페이스 : RS-232C 시리얼, 센트로닉스 패러렐
- 컬럼수 : 136
- 소비전력 : 140W
- 중량 : 14.5kg
- 추가(선택) : 낱장공급장치, 양방향 전산용지, SHEET FEEDER, RS-232C I/F카드

**제조원 및 판매원 : 대우통신**
**전 화 : (02)771－35(교환)4133, 34**

## 포터블형 전자타자기 TQ-120



TQ-120은 한글·영문 겸용이며 2벌식 표준자판을 채택하고 있고, 오타자동수정이 용이하며, 휴대가 편리하도록 되어 있다.

또한 가격도 수동식타자기 수준인 20만원대로 보고 있으며 내수는 물론 수

출전망도 밝아 년간 30만대 규모의 수출상담이 진행중이다.

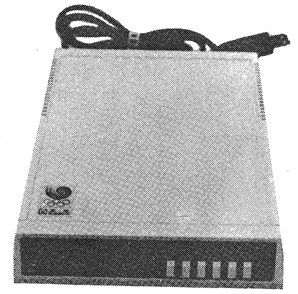
이 제품의 주요 규격을 보면 다음과 같다.

- 타자방식 : 페이지 휠방식
- SPOKE수 : 100개/프린트휠
- 타자지 폭 : 12.4인치
- 타자폭/속도/문자간격 : 10인치/10CPS/10, 12CPi
- 라인간격 : 1, 1½, 2LPi
- 키수 : 57키(44+13)
- 자동수정용량 : 1라인(영문, 한글)
- 리본카세트 : ① 코렉터블 카본리본
  ② 나일론 패브릭리본
- 수정리본 : LIFT OFF방식(카본리본)
- 크기(㎜):408(W)×370(D)×12(H)

**제조원 및 판매원 : 삼성전자**

**전 화 : (02)559－2615**

## 소형·경량화를 실현한 GSM-12F



GSM-12F AD1은 VLSI 및 마이크로프로세서를 사용한 소형 경량의 고신뢰성 데이터모뎀이다. 동 제품은 2선식 공중선로를 사용하여 600bps 또는 1200bps 비동기식 및 동기식 데이터를 전이중 방식으로 송수신하며 비동기식 방식일 경우 스타트, 스톱비트를 포함하여 8, 9, 10, 11비트 캐릭터를 송수신할 수 있다.

후면판의 "TELCO" 단자에 공중선을 연결하고, "TELSET" 단자에 전화선을 연결하면 수동호출도 가능하다. 또한 자동호출기능도 가지고 있어, 사용자가 직접 전화를 거는 불편함이 없이 상대를 호출할 수 있으며 MFC 또는 PULSE방식을 사용자의 명령어로 선택 가능하다.

그리고 자동응답기능이 가능하며 A-borttimer를 내장, 선로에 신호가 없을 경우 자동으로 선로를 차단하게 하였다.

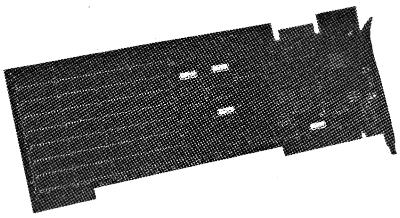
그외 특징을 간추려 보면 다음과 같다.

- 자동 재호출기능
- 전화번호 기억 및 수정기능
- 키보드 다이얼링
- 원터치 다이얼링기능
- 랜덤 다이얼링기능
- 호출상태 및 통신상태 감시기능
- 음성 통화용 전화기 범용가능
- 다양한 시험기능(TLB, DLB, RLB, ST)
- "HELP" 정보기능
- 비휘발성 메모리 사용(EEPROM)
- 110V/220V 겸용

**제조원 및 판매원 : 금성반도체**

전 화 : (02)787－2734

## CAD용 그래픽 전용카드 오로라 1024TM



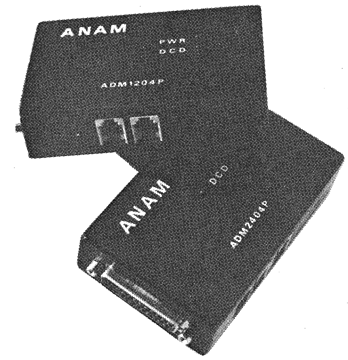
이 제품은 텍사스 인스트루먼트사의 TMS34010(50MHz) 프로세서를 탑재해 VGA, EGA카드에 비해 처리속도가 20배 이상 빠르고 1024×768의 고해상도를 지원하며 256가지의 컬러를 제공한다.

또한 IBM 3514/A 카드와 호환성이 있으며 다양한 소프트웨어를 사용할 수 있는 드라이버와 IBM의 비디오램을 사용, 고해상도를 요구하는 CAD나 그래픽 소프트웨어를 하나의 카드에서 처리할 수 있다.

**제조원 및 판매원 : 제우스컴퓨터(주)**

**전 화 : (02)784－7841**

## 초소형 휴대형 모뎀 ADM시리즈




이 두제품, 「ADM1204P」와 「ADM2404P」는 110×65×24㎜의 초소형으로 휴대가 간편한 것이 특징이다.

데이터전송속도는 「ADM1204P」인 경우 1200bps(DPSK), 300bps(FSK), 「ADM2404P」인 경우 2400bps(QAM), 1200bps(DPSK), 300bps(FSK)이며 두 모델 모두 9V건전지나 어댑터를 이용할 수 있다.

한편 이용가능한 통신용 소프트웨어는 크로스토크 16, PC-토크 Ⅲ, 그리고 스마트콤 Ⅰ과 Ⅱ 등이다.

**제조원 및 판매원 : 아남전기산업**

**전 화 : (032)524－1321**

# 마이크로소프트사 MS-DOS V4.0 발표

미국 마이크로소프트사는 지난 7월 19일 MS-DOS 버전 4.0을 발표, PC메이커들에게 OEM공급을 시작하였다. 이는 같은 시기에 IBM이 발표한 DOS 버전 4.0과 일부 유틸리티를 제외하고는 동일한 것으로써 미국내 PC메이커들은 이미 컴퓨터 사용자들에게 공급을 시작했으나 국내에서의 정상적인 출하시기는 내년부터로 보인다.

주요 강화된 내용을 보면, 우선 예전부터 항상 문제가 되어 왔던 하드디스크 용량의 상한을 32MB에서 2GB로 대폭적인 확장을 하였다. 다음에 확장메모리 드라이버 EMS 4.0을 표준장치로 최대 32MB까지의 메모리를 관리할 수 있게 되었다. 세번째로 캐릭터 베이스의 가상셀을 탑재, 마우스에 의한 조작 등을 간편하게 하고 어플리케이션 범위를 더욱 크게 하고 있다.

386시스템을 중심으로 40MB 이상의 대용량 하드디스크 탑재가 상식화된 지는 이미 오래되었다. 그러나 종래판의 DOS에서는 최대 32MB까지밖에 관리할 수 없기 때문에 이것보다 큰 용량의 디스크를 사용할 경우에는 파티션이란 영역으로 나누어 별개의 드라이브 명칭으로 이용했던 것에 비해 DOS 4.0은 2GB까지의 영역을 관리할 수 있기 때문에 화일시스템을 전면적으로 개편한 것임을 단적으로 보여주고 있다.

멀티태스크용 OS인 OS/2의 현행 버전도 종래의 MS-DOS와 기본적으로 동일한 화일 시스템을 사용하고 있기 때문에 디스크 용량이 최대 난점이지만, 머지 않아 DOS 4.0과 동일한 화일시스템을 채용할 것으로 보인다.

디스크 용량에 따른 제한범위의 변화를 **표 1**에 나타냈다. 중요한 점은 섹터수의 상한을 64K 로부터 4G로 확장시켰다는 것이며 섹터수 상한을 64K배로 하면 **표 2**와 같이 논리적으로는 최대 256T(T는 $10^{12}$) 바이트의 공간을 관리할 수 있게 된다. 그러나 실제로는 물리적인 할당의 제약이 따르기 때문에 최대 용량은 한층 적어지게 된다.

디스크의 영역은 일반적으로 클러스터라는 단위로 나누어 관리된다. 클러스터는 복수개의 섹터를 통합하기 위한 것인데 이 클러스터의 갯수는 화일 할당 테이블(FAT)로 결정되며 이전의 12비트 FAT의 경우는 최대 4K개의 클러스터를, MS-DOS 3.1 이후의 16비트 FAT에서는 64K개까지의 클러스터를 관리할 수 있다. 클러스터 길이와 FAT비트수로부터 결정되는 용량의 상한을 **표 3**에 나타냈다. 16비트 FAT를 채택한 3.1판 이후의 IBM PC용 MS-DOS 에서는 클러스터 길이가 512바이트로 고정되고 있어 16비트 FAT에서도 32MB까지 관리할 수 있게 된 것이다.

표 3으로 알 수 있는 바와 같이 DOS 4.0에서의 디스크용량 한계는 이론상 4GB인데도 2GB까지로 제한되고 있는 것은 MS-DOS 커넬이 내부적으로 화일 관리를 위하여 사용되고 있는 시스템 화일테이블(SFT)의 제한을 받고 있기 때문이다. SFT는 일반 유저들에게 공개되지 않았으므로 그 내용은 분명하지가 않다.

## 기존 화일시스템과의 호환성 문제

〔표 1〕 MS-DOS 4.0과 3.xx 이전의 디스크 제한치

| | 3.xx | 4.0 |
|---|---|---|
| 섹터 수(a) | 64K | 4G |
| 섹터 길이 (b) | 64K 바이트 | 64K 바이트 |
| 바이트어드레스(a x b) | 4G 바이트 | 256T 바이트 |
| 클러스터 수 | 4K / 64K* | 64K |
| SFT 바이트어드레스 | 불명확 | 2G 바이트 |

*4K는 12비트 FAT를 이용한 경우, 64K는 16비트 FAT를 이용한 경우

〔표 2〕 섹터길이와 어드레싱이 가능한 디스크 용량
(실제의 용량은 물리적인 할당으로 인해 제한된다)

| 섹터 길이 | 가능 용량의 상한 | |
|---|---|---|
| | 3.xx(64섹터) | 4.0(4G섹터) |
| 128바이트 | 8M 바이트 | 512G 바이트 |
| 256 | 16M | 1T |
| 512 | [32M] | 2T |
| 1K | 64M | 4T |
| 2K | 128M | 8T |
| 4K | 256M | 16T |
| 8K | 512M | 32T |
| 16K | 1G | 64T |
| 32K | 2G | 128T |
| 64K | 4G | 256T |

*사각형으로 표시된 부분이 IBM PC용 MS-DOS의 상한치이다.

〔표 3〕 클러스터길이와 디스크용량
512바이트 길이의 클러스터를 사용하고 있는 IBM PC용 MS-DOS 3.xx에는 16비트 FAT를 사용하여 32MB의 용량을 실현시키고 있다(사각형 표시부분).

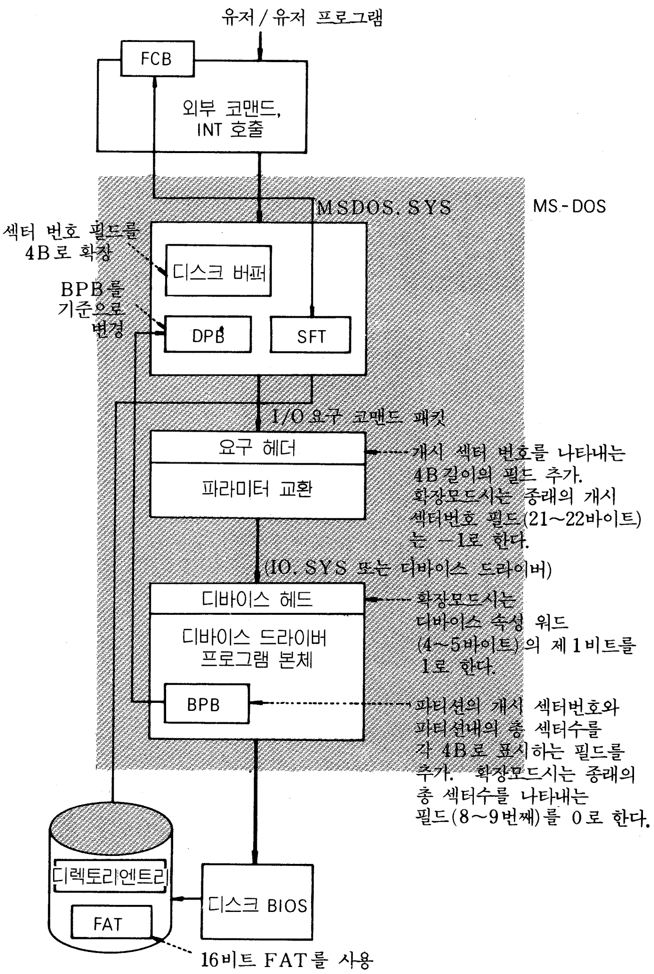
| 클러스터 길이 | 가능 용량의 상한 | |
|---|---|---|
| | 12비트FAT(4K클러스터) | 16비트FAT(64K클러스터) |
| 512바이트 | 2M 바이트 | [32M]바이트 |
| 1K | 4M | 64M |
| 2K | 8M | 128M |
| 4K | 16M | 256M |
| 8K | 32M | 512M |
| 16K | 64M | 1G |
| 32K | 128M | 2G |
| 64K | 256M | 4G |

커넬(MDOS. SYS와 IO. SYS)의 구성요소간에 취할 수 있는 파라미터 블록 중의 섹터번호 필드를 4B로 확정하고 있다(실제로는 새로운 필드를 추가하고 있다. **그림 1** 참조). 특히 종래의 화일시스템과의 호환성을 유지하기 위해서 기존방식이 확장방식인가를 판별하는 새로운 필드를 도입하고 있다.

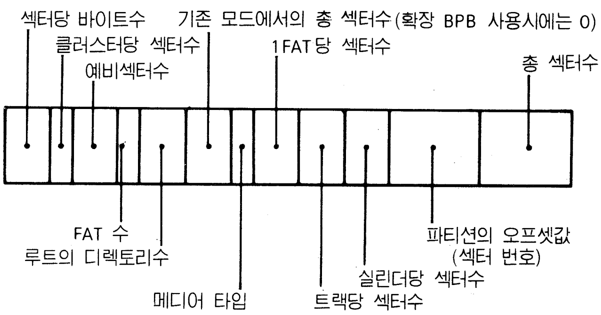
일예로써 IO. SYS(또는 디바이스 드라이버)가 관리하는 하드디스크용 바이오스 파라미터 블록(BPB)을 **그림 2**에 나타내었다. 32MB 이상의 파티션을 사용하는 경우에는 종래판에서 총 섹터수를 지정하고 있는 2바이트 필드값을 0로 하여 말미에 추가한 2개의 필드(각 4바이트)를 사용한다.

MS-DOS 4.0에서는 3.xx 판 이후에 작성된 화일도 액세스할 수 있는데, 반대로 4.0판으로 작성한 32MB 이상의 파티션에 있는 화일을 종래의 버전에서는 액세스할 수 없다는 점이 MS-DOS 4.0의 커다란 결점으로 지적되고 있다.

**〔그림 1〕 MS-DOS 디스크 제어구조와 버전 4.0에서 변경된 부분**




**〔그림 2〕 확장 BPB(BIOS PARAMETER BLOCK)의 구조**




## 1MB를 초월한 영역에서도 이용

LIM EMS 4.0 기준의 확장메모리 드라이버를 표준장치로 하고 MS-DOS가 관리하는 주기억영역을 초월한 메모리(최대 32MB) 용량도 이용할 수 있게 되었다.

또한, 마이크로소프트사는 EMS 4.0에 추가하여, 1MB를 초과한 영역을 이용할 수 있는 XMS을 MS-DOS의 메모리관리 표준방식으로서 계속 추진중에 있다고 한다.

이 XMS에서는 우선, MS-DOS가 관리하는 640 KB의 주기억영역에 64KB를 추가할 수 있고(HMA), 또 HMA보다 상위 공간을 데이터 기록영역(EMB)으로서 이용할 수도 있다. 여기서 HMA는 1024K~1088KB 사이에 위치하는 64KB의 영역으로써 보통 디바이스 드라이버나 상주 종료형(TSR) 프로그램을 이곳에 위치시킨다.

## 마우스와 풀다운 메뉴로 조작할 수 있는 가상셀

또한 4.0에서는 캐릭터 베이스의 윈도우 환경(DOS SHELL)을 표준으로 제공, 화일일람 표시나 어플리케이션의 기동 등 기본적인 DOS기능을 풀다운 메뉴와 마우스로 조작할 수 있으며 종래의 코맨드형 방식과 셀환경을 유저가 선택할 수 있게 되었다. IBM의 DOS 4.0도 대개 유사한 조작 환경을 갖는 DOS 셀을 탑재하고 있다.

**〔용어해설〕**

- **BIOS** : basic input/output system
- **BPB** : BIOS parameter block
- **DPB** : device parameter block
- **EMS** : expanded memory specification
- **FAT** : file allocation table
- **FCB** : file control block
- **HMA** : high memory area
- **LIM** : Lotus/Intel/Microsoft
- **SFT** : system file table
- **TSR** : terminate and stay resident
- **XMS** : extended memory specification

# 구입가이드

## 본체 XT 명필 XT

● 제조원 : 고려씨스템(주)
● CPU : Intel 8088-Ⅱ(8087 코프로세서 옵션)
● 클럭 속도 : 8MHz
● 주기억용량 RAM : 512KB/ROM : 16KB
● 보조기억장치 FDD : 500KB(5.25인치)
HDD : 20MB(5.25인치)
● 확장 슬롯 : 8개(풀사이즈 6개, 하프사이즈 2개)
● 통신 포트 직렬 : RS-232C 2개
병렬 : 센트로닉스 규격 1개
● OS : MS-DOS
● 가격 : 180만원(1HDD, 1FDD)
150만원 (2FDD)
● 문의처 : 718-4411

## 본체 XT 마이티-16Ⅲ

● 제조원 : 금성사
● CPU : Intel 8088 Ⅰ(8087 코프로세서 옵션)
● 클럭 속도 : 4.77/10MHz
● 주기억용량 RAM : 640KB
● 보조기억장치 FDD : 360KB×1
HDD : 20MB
● 통신 포트 직렬 : RS-232C 1개/
병렬 : 1개
● OS : MS-DOS
● 가격 : 129만원
● 기타 특성 : 중·대형 컴퓨터와 접속하여 한글 터미널 기능 수행. 토큰 링 방식으로 구축. 고집적 커스텀 IC 채용.
● 문의처 : (02)701-1114(대)

## 본체 XT CPC-4000L



● 제조원 : 대우전자
● CPU : Intel 8088
● 클럭 속도 : 4.77MHz
● 주기억용량 RAM : 256KB(최대 640KB)
ROM : 32KB
● 보조기억장치 FDD : 500KB×1(5.25인치)
HDD : 20MB(5.25인치 옵션)
● 확장 슬롯 : 4개
● 통신 포트 병렬 : 1개
● OS : MS-DOS
● 가격 : 69만원
● 기타 특성 : 칼라 모드가 내장되어 있어 영문 모드에서 16칼라 모드의 변환이 쉽다.
● 문의처 : (02)759-5637

## 본체 XT PRO-2000

● 제조원 : 대우통신
● CPU : Intel 8088-Ⅱ(8087-Ⅱ 코프로세서 옵션)
● 클럭 속도 : 4/8MHz
● 주기억용량 RAM : 512~640KB
ROM : 16/32KB
● 보조기억장치 FDD : 360KB×2(5.25인치)
HDD : 20MB
● 통신 포트 직렬 : 1개
병렬 : 1개
● OS : MS-DOS
● 가격 : 135만원(2FDD)
192만원(1HDD, 1FDD)
● 기타 특성 : VLSI를 채택. 원보드상에 다양한 기능과 설계를 최적화. IBM 2바이트 조합형과 정부 표준 완성형 코드 및 7비트, 2바이트 완성형 코드를 선택할 수 있다.
● 문의처 : (02)771-35(교환 7777)

## 본체 XT ROMAX-$V_2$



● 제조원 : 로얄컴퓨터
● CPU : NEC(V20)
● 주기억용량 RAM : 640KB/ROM : 64KB
● 보조기억장치 FDD : 0.5MB×2(5.25인치)
● 확장 슬롯 : 8개
● 통신 포트 직렬 : 1개(옵션)

병렬 : 1개(옵션)
●OS : MS-DOS
●가격 : 98만원
●기타 특성 : 가격대 성능비가 우수하며, 회로의 안정성이 우수해 주변환경이나 조건에 영향받지 않고 사용할 수 있다.
●문의처 : (02) 785-3458

## 본체 XT 트라이젬 POP

●제조원 : (주)삼보컴퓨터
●CPU : Intel 8088-I
●클럭 속도 : 10MHz
●주기억용량 RAM : 640KB
●보조기억장치 FDD : 360KB×2(5.25인치)
HDD : 20MB
●확장 슬롯 : 5개
●통신 포트 직렬 : RS-232C 1개
병렬 : 1개
●OS : MS-DOS
●가격 : 129만원(2FDD)
174만원(1FDD, 20MB HDD)
●기타 특성 : 인텔리전트 한글 터미널 기능을 완벽히 지원하며, 상하위 PC기종과 LAN시스템 구성하여 워크스테이션 기능 보유
●문의처 : (02) 3902-114

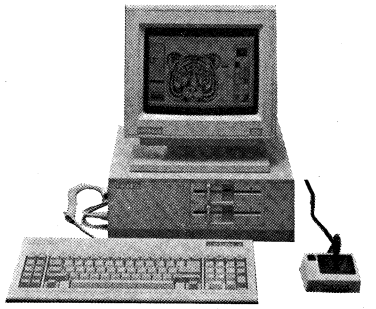
## 본체 XT SPC-3000B

●제조원 : 삼성전자
●CPU : Intel 8088-I
●클럭 속도 : 10MHz
●주기억용량 RAM : 512KB/ROM : 32KB
●보조기억장치 FDD : 500KB(5.25인치)
HDD : 20MB(5.25인치)
●통신 포트 직렬 : RS-232C
병렬 : 프린터 I/F
●OS : MS-DOS
●가격 : SPC-3000B SG(1FDD)는 132만원
SPC-3000B 2G(1FDD, 20MB HDD)는 186만원
●기타 특성 : 중·대형 컴퓨터의 워크스테이션으로 한글 처리가 가능한 상태로 연결하여 사용할 수 있다.
●문의처 : (02) 559-2643

## 본체 XT 포커스 100

●제조원 : 일진전자
●CPU : Intel 8088-II
●클럭 속도 : 8MHz
●주기억용량 RAM : 512KB
●보조기억장치 FDD : 360KB(5.25인치)
HDD : 10MB(5.25인치)
●확장 슬롯 : 5개
●통신 포트 직렬 : 2개/병렬 : 1개
●OS : MS-DOS
●가격 : 155만원
●기타 특성 : 각종 보드를 내장하여 고장율을 감소시키고, 한글 한자코드를 국가 표준안에 적용.
●문의처 : (02) 888-6813~4

## 본체 XT 슈퍼-16E



●제조원 : 현대전자
●CPU : Intel 8088-II
●클럭 속도 : 4.77MHz
●주기억용량 RAM : 256KB(최대 512KB)
●보조기억장치 FDD : 360KB×1(5.25인치)
●확장 슬롯 : 5개
●통신 포트 병렬 : 1개
●OS : MS-DOS
●가격 : A모델 85만원/B모델 74만원
●기타 특성 : 컴퓨터 교육과 학습 생활·업무 이용이 가능. 정부 표준 워크스테이션과 동일한 한글 키보드 채택.
●문의처 : (02) 741-3316

## 본체 XT Hope-16000S



●제조원 : 희망전자
●CPU : NEC V40
●클럭 속도 : 5MHz/8MHz
●주기억용량 RAM : 256KB
●보조기억장치 FDD : 360KB(5.25인치)
●확장 슬롯 : 4개(EGA, ADC, HDC 내장)
●통신 포트 직렬 : RS-232C 1개
병렬 : 센트로닉스 규격 1개
●OS : MS-DOS, CP/M-86, CP/M-80
●가격 : B형(1FDD) 70만원,
C형(2FDD) 85만원
●기타 특성 : 허큘리스, CGA, MDA 기능을 내장하여 칼라, 모노의 자유로운 선택 가능.
●문의처 : (02) 275-2888

**본체 XT QUE-XT Turbo**

- 제조원 : 북악시스템
- CPU : Intel 8088-Ⅱ
- 클럭 속도 : 8MHz
- 주기억용량 RAM : 640KB/ROM : 8KB
- 보조기억장치 FDD : 360KB×2(5.25인치)
- 확장 슬롯 : 8개
- 통신 포트 직렬 : 1개/병렬 : 1개
- OS : MS-DOS
- 가격 : 80만원
- 문의처 : (02) 266-2444

**본체 XT DELTA XT**

- 제조원 : 영동컴퓨터시스템
- CPU : Intel 8088-Ⅱ
- 클럭 속도 : 10MHz
- 주기억용량 RAM : 640KB/ROM : 48KB

# 시스템 사용시 주의점

앞으로 다가올 사회가 어떠한 것이냐를 대변이라도 해주듯 컴퓨터의 보급은 갈수록 늘고 있다. 특히 16비트급에서 그 추세는 두드러지며 특수용도를 위한 전문가들이 아닌 일반 사용자들에게는 IBM PC XT 또는 호환기종들이 이미 익숙해져 있는 실정이고 인텔의 80286과 80386 마이크로프로세서를 탑재한 기종들도 대거등장, 컴퓨터(PC 수준으로 표기함)대중화의 길은 숨가쁘기만 하다.

어떤 종류의 문명기기라도 이용도가 증가할수록 그만큼 자주 발생하는 고장 및 이와 유사한 문제로 고민을 하게 되는 빈도는 늘어난다. 그래서인지 요즈음의 컴퓨터 판매광고를 보면 아프터서비스AS 부문을 강조하지 않는 업체는 찾아보기가 힘들다.

여기서 한가지 짚고 넘어가야 할 것은 아직도 일반 가전제품과 컴퓨터를 동일 선상에서 볼수 없는 것이(가격적인 면을 떠나서) 컴퓨터를 어느 정도 이해하느냐에 따라 이러한 아프터서비스의 의미는 크게 달라지게 된다. 비근한 예로, TV나 냉장고를 사용하다 고장이 발생한 경우 아프터서비스의 도움을 기다리거나 관련 업소로 운반해서 수리 또는 부품을 교체하는 등의 현상과 컴퓨터시스템에 문제가 발생해서 수리를 하고 부품을 교체하는 등의 현상은 그 의미가 사뭇 다를 수가 있는 것이다.

컴퓨터 관계의 대부분의 아프터서비스는 판매된 하드웨어에서 사용할 수 있는 양질의 소프트웨어를 개발, 지원하고 이들의 효율적인 운영을 위한 교육적인 부분으로 이루어진다. 즉 이런 측면이 아닌, 시스템 자체에 문제가 자주 발생하여 아프터서비스를 받은 것은 별로 기분좋은 일이 못되며 그러한 제품을 가진 사람은 구입에 있어 꽤나 운이 없는(?) 케이스에 속하게 되는 것이다. 바로 이런 문제들 때문에 컴퓨터 구입시는 가격과 안정성이 늘 고민의 대상이 되곤 한다.

그러나 컴퓨터의 경우 이렇게 수리를 해야 하는 등의 상황은 대체로 드물게 나타나며 올바르게 이용을 했을 때 엄청난 기능과 업무의 효율을 얻을 수 있는 게 보통이다.

컴퓨터를 이용하다가 조그만 이상이 생기면 시스템의 질과 안정성을 먼저 의심하여 아프터서비스를 요청하기 쉬운데 알고보면 특별한 경우를 제외하고 시스템구성 및 미숙한 조작에 의한 경우가 많으므로 이를 하나하나 이해하다보면 불필요한 고민과 비용을 절감할 수 있으며 또한 이런 종류의 시각으로도 한번쯤 보는 것이 컴퓨터를 이해하는데 유익한 도움이 되리라 본다. 특히 앞으로 본지에 게재될 Q/A란도 이러한 문제들을 독자들과 같이 해결하는데 많은 도움이 됐으면 하는 바램이다.

## 1. 시스템 보드

이 부분은 컴퓨터시스템의 핵심부로서 사람에 비유하면 두뇌와 신경조직을 감싸주고 있는 머리부분과 심장등을 감싸주는 몸통부분에 해당된다. 이 외에 손과 발, 그리고 각종 도구에 해당하는 장치들이 연결되어 하나의 시스템을 구성하게 되는 것이다.

보통 '컴퓨터 본체'라고 부르는 곳이 여기이며 마이크로프로세서(CPU), 주기억장치(ROM,RAM), 확장 슬롯, 그외 몇 개의 코넥터류와 부속구성부로서 전원과 스피커로 이루어져 있다.

이 부분에 이상이 있게 되면 치명적이 될 수 있으며 대개는 각종 연결부위의 오류 또는 제품 사양에 어긋나는 부품연결, 원하는 기능부품의 부재등으로 인한 오류등이 대부분이다.

**• 전원공급장치**

시스템 보드와 키보드에 직류전원을 공급하는 기능을 가지며 +5, −5, +12, −12V를 출력한다. 이 부분에 이상이 있다고 판단되는 경우는 먼저 전원코드가 제대로 접속되어 있는지를 체크한다. 보통 자주 휘어지는 부위의 플러그쪽에서 선이 끊어지는 수가 있다. 그리고 본체의 온도가 너무 높을 때는 회전 팬이 제대로 동작하고 있는지 조사하고, 정전이 심한 곳이나 전압변동이 심한 곳에서 컴퓨터를 사용 하려면 무정전 전원공급장치(UPR)나 자동 전압조정장치(AVR)를 연결해서 쓴다.

**• 마이크로프로세서**

여기에서 중심으로 설명되는 IBM PC XT시스템의 CPU는 1979년 인텔사가 발표한 8088을 쓴다.

이 외에 내부 IO 버스 구조가 조금 다른 8086, NEC V40

●보조기억장치 FDD : 0.5MB×2(5.25인치)
●확장 슬롯 : 8개
●OS : MS-DOS, PC DOS, CP/M-86, Venix
●가격 : 75만원
●문의처 : (02)713-9528

## 본체 XT ASI-800T

●제조원 : 에이스컴퓨터
●CPU : Intel 8088-I
●클럭 속도 : 10MHz
●주기억용량 RAM : 256KB
●보조기억장치 FDD : 360KB×2(5.25인치)
●확장 슬롯 : 8개
●OS : MS-DOS
●가격 : 70만원
●문의처 : (02)273-2685

등을 탑재해 쓰고 있는 종류도 있으나 국내에서 MS-DOS를 표준 OS로 채택 하고 있는 XT급 PC중 90% 이상이 8088계열이다.

그밖에 1984년에 발표한 PC-AT 용으로는 80286프로세서가 쓰이고 있으며 32비트에서는 80386프로세서를, 그리고 매킨토시같은 경우는 모토롤라사의 68000프로세서를 탑재하고 있다. 이와같이 여러 CPU가 쓰이고 있으므로 구입시 신중한 판단을 해야 한다. 각 CPU마다의 특징들로 인한 문제점을 시스템의 고장으로 잘못 오인되서는 안되기 때문이다.

**• 속도**

CPU의 처리속도도 날로 빨라지고 있다. 각 업체끼리 경쟁이라도 하듯 기존의 4.77MHz와 8MHz 옵션의 속도에서 클럭주파수와 주변 칩을 개조, 급기야는 10MHz까지 기능하게 만들고 있다. 속도가 빠를수록 대량의 데이터를 신속하게 처리할 수 있으며 사용상의 지루함을 줄일 수는 있겠으나 그렇다고 무조건 빠른 것만이 좋은것은 아니다. 정상적 처리속도 실현은 프로세서 뿐만 아니라 그와 연관되는 주위의 칩들도 그 속도에 충분히 부응해 주어야 하기 때문에 너무 무리한 속도를 요구하면 도리어 시스템에 문제를 안겨줄 수 있다. 보통 시스템 보드상에 있는 딥스위치를 조절하여 선택한다. 종합하여 말하면 프로세서 제조시 표준으로 결정된 속도가 제일 안정하며 제품 구입시 속도에 너무 치우쳐 시스템의 안정성을 간과하는 오류를 범하지 않기 바란다.

**• 확장 슬롯**

확장 슬롯은 슬롯 하나하나에서 나오고 있는 62개의 접점에 의해서 컴퓨터는 모니터, 프린터, 확장 메모리, 모뎀, 디스크등의 주변기기를 사용할 수 있게 한다. 이 부분도 제대로 이해하지 못했을 때 억울하게 컴퓨터가 고장으로 오인받기 쉬운 곳으로 알맞은 그래픽 카드를 장만하지도 않고 남들이 다 즐기는 그래픽을 자기만 안된다고 불평을 하거나, 한글이 제대로 나오지 않다거나, 또는 동일모니터에서 그래픽해상도가 틀리는 경우, 프린터를 연결하고 싶을 때, 조이스틱등을 연결하고 싶을 때, 시리얼포트를 이용한 통신을 하고 싶은데 안되는 경우등은 일단 보유하고 있는 컴퓨터의 기종과 올바른 콘트롤러 또는 어댑터카드들을 체크해 보는 것이 옳다. 구입전에 가능한 기능들과 연결 가능한 옵션 카드 및 장치들에 대해 상담을 하는 것도 좋은 방법이다. 이런 종류의 부품들은 대부분 옵션으로 필요로 할 때 구입하는 것으로 컴퓨터를 여러 분야로 활용하고 싶을 경우 중요한 부분이다. 슬롯의 갯수 및 접촉불량도 체크해 본다. 한글처리문제도 허큘레스 보드와 한글보드방식을 사용할 때 틀린 부분이 많으므로 잘 선택해야 하며 갈수록 한글보드방식이 늘어나고 있는 추세이며 소프트웨어도 이에 맞추어 개발되고 있다.

## 2. 키보드

키보드는 대표적인 입력기기로서 예전에는 본체와 붙어 있었으나 16비트 PC가 등장하면서 분리형으로 나오고 있다. 대표적인 종류로는 84키형과 101키형이 있다. 그리고 이 부분에 자주 나타나는 고장 및 문제점을 보면 키를 눌렀는데도 해당 신호가 전달되지 않는 경우가 대부분이다. 이 경우는키를 너무 무리하게 다루어 접촉부분에 이상이 생기거나(특히 게임프로그램을 키보드로 행하는 것은 금지사항이다. 이런 경우는 조이스틱등을 쓴다.) 또는 먼지나 머리카락, 불순물등이 키사이로 들어가 기판의 접촉을 중첩되게 하여 신호전달에 혼란을 야기시키는 상태이다. 또 한가지 분리형 키보드일 때 무리하게 당기거나 자주 옮기는 경우로써 본체와 연결되는 코넥터(키보드 IO)부위가 손상되거나 헐거워짐으로써 생기는 문제들이 있다. 될 수 있는 한 키보드를 고정시켜 사용하고 시스템 주위를 깨끗히 하여 먼지등을 적게 하라는 것은 새삼 강조할 필요가 없는 줄로 안다. 그런데 한가지 알아두어야 할 사항은 소프트웨어를 실행하다 도중에 키입력이 되지 않는 경우가 있는데 이런 경우는 키보드에 문제가 있는 것이 아니라 프로그램 실행도중 잘못된 제어키들을 입력해진 경우가 많은데 이 경우는 고장이 아니므로 재부트시키면 정상적으로 구동된다.

## 본체 XT TOP 88

●제조원 : 하나컴퓨터
●CPU : Intel 8088-I
●클럭 속도 : 10MHz
●주기억용량 RAM : 512KB/ROM : 8KB
●보조기억장치 FDD : 360KB×2(5.25인치)
●확장 슬롯 : 8개
●통신 포트 직렬 : 1개/병렬 : 1개
●OS : MS-DOS
●가격 : 90만원
●기타 특성 : 한글 한자 기능을 원보드시스템으로 EGA 텍스트상에서 한글 2,770자, 한자 4,888자, 유저 정의문자 1,024자를 지원
●문의처 : (02)702-5579

## 본체 XT 프로스타 TURBO XT

●제조원 : 한국퍼스컴
●CPU : Intel 8088-I
●클럭 속도 : 10MHz
●주기억용량 RAM : 640KB/ROM : 64KB
●보조기억장치 FDD : 360KB×2(5.25인치)
●확장 슬롯 : 8개
●OS : MS-DOS
●가격 : 82만원
●문의처 : (02)701-1655~6

## 본체 XT VT-16+



●제조원 : 유니온컴퓨터
●CPU : Intel 8088-I
●클럭 속도 : 10MHz
●주기억용량 RAM : 640KB/ROM : 32KB
●보조기억장치 FDD : 0.5MB×2(5.25인치)
●확장 슬롯 : 8개
●OS : MS-DOS
●가격 : 96만원
●기타 특성 : CAD 시스템 가동시, 그래픽 처리 및 포트란 등 과학계산용을 빠르고 신속히 처리.
●문의처 : (02)587-3500

## 본체 AT NANO 286A-10



# 하드디스크

## 고속대용량 기억장치

처음에 이 세상에 나타난 하드디스크는 1956년에 발표된 IBM350 자기디스크장치이다. 24인치의 디스크 50매를 매초 20회전시켜서 기억용량은 겨우 5MB, 시이크시간도 평균0.6초였다.

하드디스크에서 자주 귀에 접하는 말로「윈체스터형」이라는 것이 있다. 윈체스터란 1979년에 발표된 IBM3340의 개발코드이름이며, 이때 발표되었다.

- 디스크와 헤드를 방진용기에 넣는다.
- 디스크의 회전에서 헤드를 띄운다고 하는 기술을 사용한 것을 윈체스터형이라고 하고 있다. 이것에 의해 1978년에는 디스크의 직경을 8인치로 한 것이, 1980년에는 5인치의 디스크를 사용한 것이 발표되어 하드디스크의 소형화에 연결되었다. 최근에는 3.5인치의 것이 주류로 되어 있다.

## SCSI 인터페이스

SCSI(Small Computer System Interface: 스카시라고 불리우고 있다)란 최근 각광을 받고 있는 소형컴퓨터용의 I/O인터페이스이다. 이 인터페이스는 문자대로 소형컴퓨터용으로 구성된 것이다. 원래는 미국슈가트사가 생각해 낸 퍼스컴용 자기디스크장치 인터페이스 SASI(Shugart Assosiates System Interface)를 기초로서 더욱 범용성을 가한 것이다. 데이터전송은 바이트 단위(8비트)밖에 마련되어 있지 않다. 그렇다고는 하지만 현재에는 ANSI X3.131-1986으로서 정리되어 있어서 최고 4M바이트/초의 전송속도(동기전송시, 비동기로는 최고 1.5M바이트/초)로 되어 있다.

하나의 SCSI버스에 동시에 접속할 수 있는 장치(SCSI장치)의 수는 최대8대이다. 이 SCSI장치란 CPU나 하드디스크 등의 것이지만 단지 그들에 머무르지 않고 SCSI에 접속가능한 것이면 무엇이라도 되는 것이다.

SCSI와 나란히 자주 눈에 띄는 말에 ESDI(Enhanced Small Device Interface)라는 것이 있다. 이것은 SCSI인터페이스와 실제의 하드디스크 드라이브와의 사이의 인터페이스로서 잘 사용되고 있다.

또 계측기 등의 접속에 종래부터 사용되고 있는 GPIB(General Purpose Interface Bus)라는 것도 있다. 이것도 8비트의 패러럴데이터전송이며 옛날에는 하드디스크와의 접속에 사용된 것도 있다.

## 사용상의 주의

### • 진동과 충격

하드디스크라는 것은 충격에 약한 것이다. 특히 동작중인 고속으로 회전하는 디스크와 헤드의 간격은 담배연기보다 작은 것이며, 예를 들면 정보젯트기가 고도 5m로 날고 있는 것 같은 것이다. 따라서 진동이 심한 프린터나 덜거덕거리는 책상위 등에서 사용하는 것은 피하는 쪽이 좋을 것이다.

하드디스크를 이동시킬 때는 조용하게 한다. 또 운송 등은 메이커지정의 포장으로 하도록 한다. 차에서 덜거덕거리게 이동하는 것은 위험하다.

### • 외부자계

하드디스크는 의외로 외부의 자계 영향을 받는다. 특히 전자쉴드가 불완전한 모니터 등의 바로 옆에서는 판독에러가 발생하기 쉽게 된다. 하드디스크측에도 자기쉴드가 시행되어 있는 것이 요망되고 있다. 모니터나 텔레비젼은 의외로 강한 자계를 발생하고 있는 것이 있으므로 될수록 멀리 놓는 쪽이 좋을 것이다.

### • 먼지

하드디스크의 심장부는 먼지를 싫어한다. 그러나 디스크의 원반이나 헤드가 들어있는 용기는 완전밀폐로 되어 있으므로 통상의 사용에서는 문제없다. 그러나 헤드와 자기원반과의 간격은 담배연기보다 작은 것이다. 주의해서 나쁠것은 없다.

### • 노이즈

하드디스크드라이브에서 취급하고 있는 신호는 매우 미소한 것이다. 따라서 전기적인 노이즈는 판독에러의 원인으로 된다. 또 주변회로는 컴퓨터와 같든가 그것 이상으로 복잡한 것이므로 폭주할 가능성도 있다. 어느 것으로 하여도 컴퓨터에 사용하는 것과 같든가 그것 이상의 주의를 기울여야 한다. 컴퓨터 본체에서 나오는 전원노이즈가 영향을 미칠 경우도 있다. 프레임그라운드단자는 확실하게 어스하도록 한다.

### ● 12비트 FAT

MS-DOS의 화일관리의 단위로서는 화일, 클러스터, 섹터의 3개가 있다. 즉 유저에게서 볼 수 있는 것은 화일뿐이지만 하나의 화일은 실은 몇개인가의 클러스터로 되어 있다. 하나의 클러스터는 더욱 몇개인가의 섹터로 되어 있다.

MS-DOS 3.0이하에서는 하드디스크는 1섹터=1024바이트, 1클러스터=8섹터로 결정되어 있다. 또 사용하고 있지 않는 클러스터와 그렇지 않은(사용중의) 클러스터와를 관리하는 것에 FAT라는 것이 있지만 이것의 최대값은 4096(12비트)이다. 이 때문에 디스크의 용량은 1024×8×4096 으로 32MB로 된다.

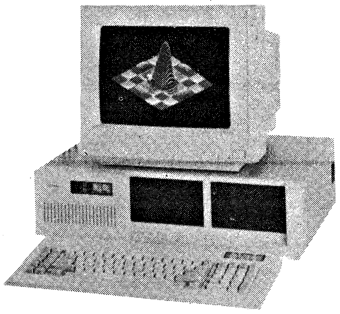
V3.1의 경우 FAT의 최대가 65536(16비트)으로 고쳐졌으므로 최대용량은 크게 증가한다. 또 1클러스터당의 섹터수도 크게 늘었다.

●제조원 : 제일나노텍
●CPU : 80286-10
●클럭 속도 : 8/10MHz
●주기억용량 : ROM : 32KB
RAM : 512KB
●보조기억장치 FDD : 5.25″×1(1.2MB)
●확장 슬롯 : 8개
●통신 포트 직렬 : 1개/병렬 : 1개
●가격 : 2백10만원
●문의처 : (02) 782-3345~8

## 본체 AT SAM-AT

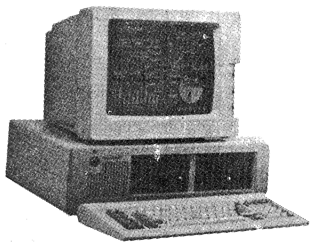
●제조원 : (주) SAM KOREA
●CPU : i80286
●클럭 속도 : 10MHz
●주기억용량 ROM : 64KB/RAM : 1MB
●보조기억장치 FDD : 5.25″×1(1.2MB)
HDD : 20MB×1
●확장 슬롯 : 8개
●통신 포트 직렬 : RS-232C
●OS : MS-DOS, PC-DOS, CP/M-86, XENIX, UCSD-P
●가격 : 2백만원
●문의처 : (02) 782-2665~6

## 본체 AT SUPER-286



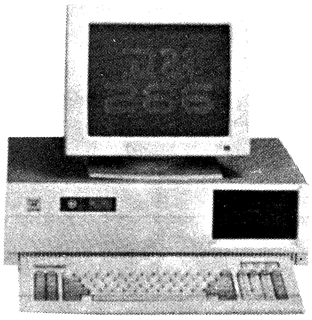
●제조원 : 현대전자
●CPU : i80286
●클럭 속도 : 10MHz
●주기억용량 RAM : 512KB
●보조기억장치 FDD : 5.25″×1(1.2MB)
HDD : 20MB
●확장 슬롯 : 12개
●가격 : 1백89만원
●문의처 : (02) 741-1311/25

## 본체 AT AT/3620



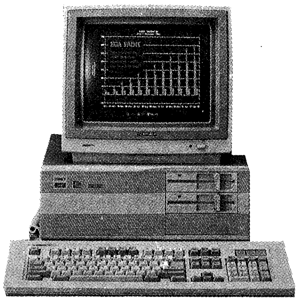
●제조원 : 대우전자
●CPU : i80286
●클럭 속도 : 8MHz
●주기억용량 RAM : 640KB
●보조기억장치 FDD : 1.2MB/HDD : 20MB
●OS : MS-DOS 3.3, XENIX-V
●문의처 : (02) 771-91, 92

## 본체 AT 고려 286



●제조원 : 고려시스템(주)
●CPU : i80286-8
●클럭 속도 : 8MHz
●주기억용량 ROM : 64KB/RAM : 512KB
●보조기억장치 FDD : 5.25″(1.2MB)
HDD : 20MB
●확장 슬롯 : 11개
●통신 포트 직렬 : RS-232C
병렬 : 센트로닉스(25핀)
●OS : MS-DOS, XENIX
●가격 : 3백65만원
●문의처 : (02) 718-4411

## 본체 AT HOST-1600A



●제조원 : 호산컴퓨터
●CPU : i80286
●클럭 속도 : 8MHz
●주기억용량 RAM : 1MB
●보조기억장치 FDD : 5.25″×1(1.2MB)
HDD : 20MB
●확장 슬롯 : 8개
●통신 포트 직렬 : RS-232C
●OS : MS-DOS, CP/M-86, XENIX
●가격 : 1백85만원
●문의처 : (02) 274-5934

## 본체 AT QUE-AT Turbo

●제조원 : 북악시스템
●CPU : i80286-12
●클럭 속도 : 12MHz
●주기억용량 ROM : 32KB/RAM : 1MB
●보조기억장치 FDD : 1.2MB×1
HDD : 20MB
●확장 슬롯 : 8개
●통신 포트 직렬 : 1개/병렬 : 1개
●가격 : 2백15만원
●문의처 : (02) 782-6095

# 플로피디스크

## 플로피디스크의 역사

1972년에 미국의 IBM사가 현재의 플로피디스크 기본으로 되어 있는 8인치 편면디스크를 사용한 IBM3740데이터엔트리시스템을 발표하고 나서 16년, 또 1976년에 미국의 슈가트사가 5¼인치의 디스켓(미니플로피)를 사용하는 플로피디스크장치(FDD)를 발표하고 나서 또 12년이나 경과하였다. 최근에는 하드디스크에 밀리고 있지만 컴퓨터 보조기억장치로서의 플로피디스크의 위치는 여전히 확고하다고 볼 수 있다.

## 기록용량의 변화

이른바 IBM포맷이라고 하는 것에도 여러가지가 있다. 1972년에 발표된 디스켓의 포맷은 8인치의 디스켓 1매당 250KB밖에 데이터를 기록할 수 없었지만 현재의 5인치 2HD 디스켓에서는 1.2MB의 데이터를 기록할 수 있도록 되어 있다.

### • 8인치, 5인치, 3.5인치, 3인치, 퀵디스크

디스켓의 크기에는 여러가지가 있으며, 자성원반의 직경이나 쟈켓(넣는 물건)의 크기로 구별하고 있다.

8인치는 문자대로 약 20㎝의 크기이지만 5인치는 약 13㎝이며, 정확하게는 5.25 또는 5¼인치라고 불리우고 있다. 3.5인치는 1980년에 소니가 발표한 것이며 512바이트/섹터, 9섹터/트랙, 70트랙/면으로 315KB의 용량이었다. 그러나 5인치드라이브와 호환인 것이 발매되게 되어 현재에는 디스켓 크기가 변화하였을 뿐이며 기록방법이나 용량 등은 5인치와 호환으로 되어 있다. 3인치는 1981년에 히다찌가 중심으로 되어 발표된 것이며 512바이트 9섹터 40트랙의 180KB이지만 현재에는 거의 사용되고 있지 않다. 퀵디스크는 현재에는 홈컴퓨터 등에 사용되고 있다.

### • 편면과 양면

디스켓의 자성면을 앞뒤 모두 사용하는 것이 양면, 한쪽밖에 사용되지 않는 것을 편면디스켓이라고 한다. 구조적으로는 같지만 편면디스켓의 일부에는 자성체가 편면밖에 보증하고 있지 않는 것도 있다. 8인치나 5인치 디스켓에서는 사이드0(이른바 앞)은 실제 물리적으로는 뒤에 해당하는 쪽이다. 3.5인치에서는 물리적인 앞이 표면에 해당한다. 또 앞과 뒤에는 트랙0의 위치가 1/12인치 벗어나고 있지만 R/W헤드도 같은 만큼 벗어나고 있으므로 문제는 없다.

### • 단밀도와 배밀도

데이터를 기록하는 밀도의 차이는 단밀도(FM : Frequency Modulation)와 배밀도(MFM : Modified FM)로 나누어진다. 8인치의 단밀도에서는 데이터 전송속도는 250K비트/초, 5인치에서는 125K비트/초이지만 배밀도에서는 각각 500K, 250K로 단밀도의 배로 되어 있다.

MFM에서는 데이터의 기록밀도가 배 일경우 기록신호의 대역이 배로 되지 않도록 데이터의 변조방식에 연구가 이루어지고 있다.

### • 사이드와 트랙

뒤인가 앞인가의 구별을 사이드라고 한다. 앞이 사이드0이고 뒤가 사이드1이다.

자성원반에는 (레코드와는 달리) 동심원 모양의 트랙이 그어져 있어서 가장자리가 트랙0, 내측이 트랙34, 39, 79, 77로 되어 있다. 트랙은 통상 1인치에 48트랙(48TPI)이지만 배트랙(2DD)이나 8인치 호환의 5인치(2HD)에서는 96TPI로 되어 있다. 또 3.5인치에서는 135TPI라고 하는 고밀도도 있다. 양면디스크에서는 앞(사이드0)과 뒤(사이드1)의 2개의 트랙을 합쳐서 실린더라고 부르고 있다.

1개의 트랙중 데이터의 리드라이트단위를 섹터라고 한다. 섹터번호는 1에서 순서대로 붙여지고 있지만 물리적으로 순번으로 되어 있다고는 한정할 수 없다. 물리적인 순번과 섹터번호가 일치하지 않을 때 스큐가 걸려 있다고 한다.

### • 포맷

8인치 이외의 아무 것도 들어 있지 않는 (블랭크) 디스켓에는 여하한 데이터도 라이트할 수 없다. 즉 새하얀 종이와 같으며 데이터를 라이트하기 위한 괘선에 상당하는 것을 그을 필요가 있다. 이것을 물리적으로 포맷한다(초기화)고 한다. 포맷함으로써 자성체에 트랙이나 섹터가 형성된다. 이때에 1섹터당의 바이트수나 섹터수, 트랙수 등이 결정된다. 또 8인치 디스켓에서도 리드에러가 나오는 등 물리적으로 파괴되었을 경우에는 물리포멧에서 다시 할 필요가 있다.

트랙이나 섹터가 되어도 소프트적으로는 아직 사용할 수 없다. CP/M이나 베이직, MS-DOS 등의 OS에 의해서도 다르지만 디스켓상 관리데이터의 초기설정을 할 필요가 있다. 이것을 논리포맷이라고 한다. 이렇게 함으로써 사용할 수 있는 디스켓으로 되는 것이다. 각 OS의 포맷프로그램은 물리포맷과 논리포맷을 (필요하다면 시스템프로그램의전송도)할 수 있도록 되어 있다.

### • 디스크의 크리닝

디스크드라이브의 크리닝은 주위에 먼지가 많은 곳이나 담배의 연기가 지독한 곳에서는 전용의 크리닝용품을 사용하여 크리닝을 할 필요가 있다. 또 디스켓에 쥬스나 커피 등을 흘렸을 때는 (정말로 중요한 데이터가 아닌 한) 단념하는 쪽이 좋을 것이다. 정말로 중요한 데이터이면 일부의 디스켓메이커에서는 디스켓의 크리닝서비스를 하고 있는 것 같으므로 거기에 가지고 가면 좋을 것이다.

**본체 AT TOP 286**




- 제조원 : 하나컴퓨터
- CPU : 80286-10
- 주기억용량 ROM : 32KB/RAM : 1MB
- 보조기억장치 FDD : 5.25″×1(1.2MB)
  HDD : 30MB
- 확장 슬롯 : 8개
- 통신 포트 직렬 : 1개/병렬 : 1개
- OS : MOS-DOS, XENIX
- 문의처 : (02)717-4894

**본체 AT FAITH AT**



- 제조원 : 민음컴퓨터
- CPU : i80286
- 클럭 속도 : 8MHz
- 주기억용량 ROM : 64KB/RAM : 1MB
- 보조기억장치 FDD : 5.25″(1.2MB)
  HDD : 20MB
- 확장 슬롯 : 8개
- 통신 포트 직렬 : RS-232C(2개)
  병렬 : 센트로닉스(1개)
- OS : MS-DOS, CP/M-86, UCSD-P, XENIX
- 가격 : 1백50만원
- 문의처 : (02)277-6151

**본체 AT S-550**

- 제조원 : 삼성전자
- CPU : i80286
- 클럭 속도 : 8MHz
- 주기억용량 ROM : 32KB/RAM : 640KB
- 보조기억장치 FDD : 5.25″/HDD : 20MB
- 통신 포트 직렬 : RS-232C(2개)
  병렬 : 센트로닉스(1개)
- OS : MS-DOS, XENIX
- 문의처 : (02)771-78

**본체 AT TriGem 286S**

- 제조원 : 삼보컴퓨터

# 모뎀

퍼스컴통신은 유저의 퍼스컴과 BBS의 호스트컴퓨터를 전화로 잇는 네트워크이다. 그래서 필요로 되는 것이 모뎀이라고 불리우는 것이다. 한마디로 말하면 유저의 퍼스컴과 전화선을 연결하는 인터페이스이다.

## 모뎀의 선택법은?

전화선에 연결하여 컴퓨터와 신호의 주고받음을 하는 것이 모뎀의 일이다. 모뎀의 경우는 전화기와 전화선의 사이에 모뎀을 접속하는 것으로 된다. 또 모뎀과 전화기의 기능을 겸해 갖고 있는 것이 모뎀내장전화이다.

음향커플러라는 것은 보통의 전화기의 수화기를 그대로 사용하여 데이터의 주고받음을 하는 것이다. 주로 300bps라고 하는 저속으로 밖에 통신할 수 없다.

가격의 기준으로 되어 있는 것이 통신속도의 차이이다. 대부분의 BBS에서는 전화회선은 300bps와 1200bps의 자동전환으로 되어 있다. bps라는 것은 통신의 속도이며 빠를 수록 모뎀 그것의 가격은 높아지지만 실제 사용하고 있을 때의 전화요금이 싸진다. 최근에는 2400bps의 모뎀도 싸져 오고 있지만 BBS에 대응하고 있지 않는 것도 많으므로 주의하자. 300/1200bps로 전2중의 것이면 안심이지만 장시간 사용하고 있어서 발열이 큰 것은 될수록 피하자.

자동발신(다이얼)을 할 수 있는 모뎀이면 BBS의 전화번호를 컴퓨터에 기억시켜 두면 귀찮은 호출을 컴퓨터에 시킬 수 있다.

### • 전2중, 반2중

보통의 전화와 마찬가지로 이쪽에서 이야기하고 있는 것이 상대에게 들림과 동시에 상대가 이야기하고 있는 것이 이쪽에 들리도록 하는 경우를 전2중이라고 한다. 또 무선통신과 같이 이쪽이 수신상태인 때에는 상대의 이야기가 들리지만 송신중에는 상대로 부터의 신호를 수취할 수 없는 것을 반2중이라고 한다. 퍼스컴통신은 특수한 BBS를 제외하고 대개 전2중이다. 또 반2중의 모뎀은 전2중에는 사용할 수 없으므로 주의하자.

### • 보오레이트

bps(Bit Per Second)란 1초간에 어느 만큼의 신호를 보내는가를 나타내는 수이다. 보오레이트(BAUD-rate, 약해서 보오)라고 하는 것도 같다고 생각해도 지장이 없을 것이다. 단, 신호의 속에는 데이터의 시작을 나타내는 스타트비트나 스톱비트, 패리티등도 포함되므로 예를 들면 1200bps, 8비트, 패리티없음, 1스톱비트의 경우는 1200÷(1+8+1)이며, 매초 120바이

● CPU : i80286
● 클럭 속도 : 8/12.5MHz
● 주기억용량 ROM : 32KB/RAM : 640KB
● 보조기억장치 FDD : 5.25″(1.2MB)
HDD : 20MB
● 확장 슬롯 : 5개
● 통신 포트 직렬 : RS-232C(1개)
병렬 : 센트로닉스(1개)
● 가격 : 4백만원
● 문의처 : (02)313-0030

## 본체 AT GC-286

● 제조원 : (주)거성
● CPU : i80286
● 클럭 속도 : 8/10/12/14MHz
● 주기억용량 RAM : 1MB
● 보조기억용량 FDD : 1.2MB×1
HDD : 20MB×1
● 가격 : 1백70만원

● 문의처 : (02)548-7345

## 본체 AT 7700

● 제조원 : 큐닉스
● CPU : i80286
● 클럭 속도 : 8MHz
● 주기억용량 RAM : 1MB
● 보조기억장치 FDD : 1.2MB×1
HDD : 20MB
● 확장 슬롯 : 8개
● 통신 포트 직렬 : 2개/병렬 : 1개
● OS : MS-DOS
● 가격 : 4백20만원
● 문의처 : (02)783-4301

## 본체 AT WALCOM

● 제조원 : 뉴텍코리아

● CPU : i80286
● 클럭 속도 : 6MHz
● 주기억용량 RAM : 1MB
● 보조기억장치 FDD : 3.5″(1.44MB)
HDD : 20MB
● 통신 포트 직렬 : RS-232C
병렬 : 센트로닉스
● 가격 : 3백63만원
● 문의처 : (02)585-3891

## 본체 386 고려 386

● 제조원 : 고려시스템
● CPU : 80386
● 클럭 속도 : 16MHz/20MHz
● 주기억용량 RAM : 2MB
● 보조기억장치 HDD : 5.25″×1(20MB)
FDD : 5.25″×1(360KB)
● I/O 포트 수 직렬 : 2개/병렬 : 1개
● 확장 슬롯 : 2개(16비트)

트의 데이터를 전화회선으로 보낼 수 있다. 전화회선을 사용하고 있는 BBS에서는 주로 300bps와 1200bps이다. 또 DDX-P 라고 하는 전용회선에서는 9600bps 나 그것 이상의 통신속도를 사용하고 있다.

● 패리티

보통의 1문자의 데이터를 나타내는데에 몇비트 사용하는가를 캐릭터길이라고 한다. 대부분의 BBS에서는 8비트이지만 세계적으로는 7비트가 주류로 되어 있다.(유명한 OS인 UNIX의 통신소프트가 7비트를 기준으로 하고 있다.) 7비트에서는 통상 짝수패리티로 한다. 패리티란 2진수 데이터"1"의 짝수/홀수로 데이터의 에러를 체크하기 위한 방법이다. 8비트에서는 보통 사용할 수 없지만(패리티 없음) 7비트통신에서는 짝수패리티(패리티 이븐)가 사용된다.

1바이트의 데이터의 끝을 나타내고, 다음의 데이터와의 간격을 유지하는 것이 스톱비트의 역할이다. 옛날은 2비트라든가 1.42비트라고 하는 값(RTTY=라디오 텔레타입의 시대)이였지만 최근의 BBS에서는 모두 1스톱비트로 되어 있다.

● 플로우콘트롤

플로우콘트롤을 하고 있으면 데이터통신의 속도가 너무 빠르고, 데이터를 수신하고 있는 측이 쓸 수 없을 때(인간이 읽기 전에 화면이 스크롤업하여 버리는 때)등에 콘트롤 S를 누르면 호스트컴퓨터가 일시 데이터의 송신을 정지하여 콘트롤 Q에서 송신을 재개시킬 수 있다. 또 반대로 호스트컴퓨터의 처리가 쓸 수 없을 때에는 호스트컴퓨터에서 콘트롤 S가 보내져 오므로 콘트롤 Q가 오기까지 데이터전송을 정지시켜 둘 필요가 있다.

대개의 퍼스컴통신에서는 플로우콘트롤 있음(X-on/off 유효)으로 설정한다.

에스케이프시퀀스

에스케이프코드(16진수 1B에 이어서 미리 정해진 문자코드를 보냄으로써 문자의 색을 바꾼다든지, 화면을 소거한다든지, 커서위치의 지정을 하는 등 가지각색인 기능을 할 수 있다.

통신조작이 잘못되어 있지 않는가? 퍼스컴측에서의 통신속도의 설정이 잘못되어 있으면 비록 모뎀의 보오레이트가 맞아 있어도 통신은 성립하지 않는다.

때때로 이상한 문자가 화면에 나온다든지 하는 것은 전화회선에 노이즈가 실려 있는 경우가 많다. 이 때에는 한번 전화를 끊고 다시 전화를 걸면 회선의 상태가 좋아지는 것이 많다.

●OS : MS-DOS, XENIX V386/286, OS/2, VP/ix(XENIX에서 MS-DOS 사용)
●가격 : 690만원
●기타 특성 : 다양한 한글처리-국가 표준코드(완성형), IBM 조합형 및 7비트, 2바이트를 함께 제공. 별도의 확장 메모리 보드없이 16MB까지의 메모리를 ON 보드상에 탑재할 수 있어 소형, 경량화 함.
문의처 : (02) 718-4418

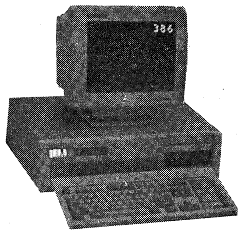
## 본체 386 GMC-8040

●제조원 : 금성사
●CPU : i80386
●클럭 속도 : 8MHz
●주기억용량 RAM : 640KB
●보조기억장치 FDD : 5.25″×1(1.2MB)
HDD : 5.25″×1(40MB)
●I/O 포트 수 직렬 : 2개/병렬 : 1개
●OS : MS-DOS, XENIX
●가격 : 736만원
●기타 특성 : 한글·한자처리 가능, 그래픽 가능
●문의처 : (02) 701-1114

## 본체 386 PRO-5000(DC-5012)

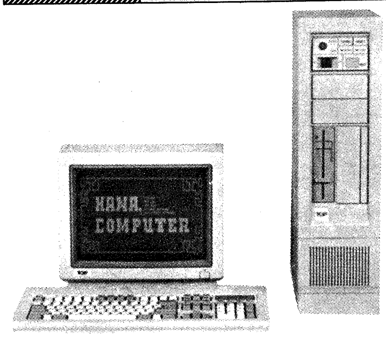
●제조원 : 대우통신
●CPU : 80386
●클럭 속도 : 8MHz
●주기억용량 ROM : 64KB/RAM : 2MB
●보조기억장치 FDD : 5.25×1(1.2MB)
HDD : 5.25″×1(40MB)
●I/O 포트 수 직렬 : 2개/병렬 : 1개
●확장 슬롯 : 6개
●OS : 영문 DOS V3.××, OS/2, XENIX-V
●가격 : 640만원
●기타 특성 : 콤팩트한 디자인과 한자(5,170자), 한글(조합형, 완성형 처리 가능)
●문의처 : (02) 771-35

## 본체 386 LEM 386



●제조원 : LEM시스템
●CPU : i80386
●클럭 속도 : 16MHz
●주기억용량 RAM : 1MB/ROM : 64KB
●보조기억장치 HDD : 5.25″×1(42MB)
FDD : 5.25″×1(1.2MB)
●I/O 포트 수 직렬 : RS-232C(2개)
병렬 : 1개
●OS : MS-DOS, OS/2, XENIX-V
●확장 슬롯 : 8개
●기타 특성 : 국산부품 비율 60%를 사용하여 제작, 뛰어난 기능의 한글/한자처리 실현.
●문의처 : (02) 567-7291

## 본체 386 TOP 386T




●제조원 : 하나컴퓨터
●CPU : i80386
●주기억용량 RAM : 2MB/ROM : 64KB
●보조기억장치 FDD : 5.25″×1(1.2MB)
●I/O 포트 수 직렬 : 2개/병렬 : 1개
●확장 슬롯 : 8개
●OS : MS-DOS, XENIX
●가격 : 394만원

●기타 특성 : 한글/한자 기능 확대(KS 표준코드), IBM PS/2 80의 VGA기능을 커버하므로 PS/2용 소프트웨어를 수정없이 사용 가능.
●문의처 : (02) 717-4894

## 본체 386 TRIGEM 386

●제조원 : 삼보컴퓨터
●CPU : i80386
●클럭 속도 : 16MHz
●주기억용량 ROM : 64KB/RAM : 2MB
●보조기억장치 HDD : 5.25″×1(40MB)
FDD : 5.25″×1(1.2MB)
●I/O 포트 수 직렬 : 1개/병렬 : 1개
●OS : MS-DOS 3.21, UNIX, XENIX
●가격 : 685만원
●기타 특성 : 고도의 엔지니어링, 연구 개발, CAD/CAM 종합 OA 구성에 최적의 기능 제공.
●문의처 : (02) 3902-114

## 본체 386 EXTRA 386



●제조원 : 삼일컴퓨터
●CPU : 80386
●클럭 속도 : 20MHz
●주기억용량 ROM : 32KB/RAM : 2MB
●보조기억장치 FDD : 5.25″×1(1.2MB)
HDD : 5.25″×1(40MB)
●확장 슬롯 : 32비트-1개, 8비트-2개, 16비트-5개
●I/O 포트 수 직렬 : RS-232C(4개)
병렬 : 1개

● OS : MS-DOS, XENIX, UNIX V
● 가격 : 430만원
● 기타 특성 : 완벽한 한글/한자 지원 모든 영문 소프트웨어에서 별도의 수정없이 한글을 구사할 수 있는 한글카드가 기본 내장되어 있다.
● 문의처 : (02) 701-6374

## 본체 32비트 Macintosh Ⅱ



● 제조원 : 엘렉스컴퓨터
● CPU : 68020
● 클럭 속도 : 15.7MHz
● 주기억용량 RAM : 1MB/ROM : 256KB
● 보조기억장치 FDD : 3.5″×1(800KB)
HDD : 5.25″×1(40MB)
● I/O 포트 수 직렬 : RS-232C, RS-422(2개)/병렬 : 1개
● OS : Mac O.S, UNIX, MS-DOS
● 가격 : 776만원
● 기타 특성 : 높은 해상도의 칼라 그래픽(256가지의 색을 선택 사용). 다목적의 4 스테레오 사운드-ASC(Apple Sound Chip) 내장. 인간과 동일한 사고방식의 하이퍼 카드 제공.
● 문의처 : (02) 783-0401

## 본체 386 UNION 386

● 제조원 : 유니온컴퓨터(주)
● CPU : i80386
● 클럭 속도 : 16MHz
● 주기억용량 RAM : 4MB/ROM : 64KB
● 보조기억장치 FDD : 5.25″×1(1.2MB)
HDD : 5.25″×1(40MB)
● I/O 포트 수 직렬 : RS-232C(1개)
병렬 : 1개

● OS : MS-DOS, XENIX, UNIX
● 가격 : 590만원
● 기타 특성 : IBM 16비트와 완전 호환기종. 16유저가 사용 가능. 고도의 엔지니어링 분야, 사무전산화. 한글, 한자 처리시 미려한 명조체 인자.
● 문의처 : (02) 587-3500

## 본체 386 ACER-1100

● 제조원 : 케이서시스템(주)
● CPU : 80386
● 클럭 속도 : 4.77/20MHz
● 주기억용량 RAM : 1MB
● 보조기억장치 FDD : 5.25″×1(1.2MB)
HDD : 5.25″×1(40MB)
● 확장 슬롯 : 7개
● OS : MS-DOS, XENIX, UNIX
● 가격 : 535만원
● 기타 특성 : 일반업무용, LAN의 화일서버.
● 문의처 : (02) 716-0101

## 본체 386 SUPER-386

● 제조원 : (주)케스타
● CPU : i80386
● 클럭 속도 : 20/24MB
● 주기억용량 RAM : 1MB
● 보조기억장치 FDD : 5.25″×1(1.6MB)
HDD : 5.25″×1(40MB)
● I/O 포트 수 직렬 : 2개/병렬 : 1개
● 확장 슬롯 : 8개
● OS : MS-DOS, UNIX(DOS 머지 OS임), XENIX 등
● 가격 : 490만원
● 기타 특성 : 미니 컴퓨터 차원을 능가하는 메모리 확장성과 CAD, 칼라 그래픽의 다양성.
● 문의처 : (02) 716-0243

## 본체 386 Tidcom-386

● 제조원 : (주)태일시스템

● CPU : i80386
● 클럭 속도 : 20MHz
● 주기억용량 ROM : 128KB/RAM : 1MB
● 보조기억장치 FDD : 5.25″×1(1.2MB)
HDD : 5.25″×1(20MB)
● I/O 포트 수 직렬 : RS-232C(2개)
● OS : MS-DOS, XENIX, UNIX
● 가격 : 630만원
● 기타 특성 : 고해상도(1024×768도트) 칼라 그래픽 사용시 한글 사용 가능, IBM-PC와 100% 호환성
● 문의처 : (02) 776-5340

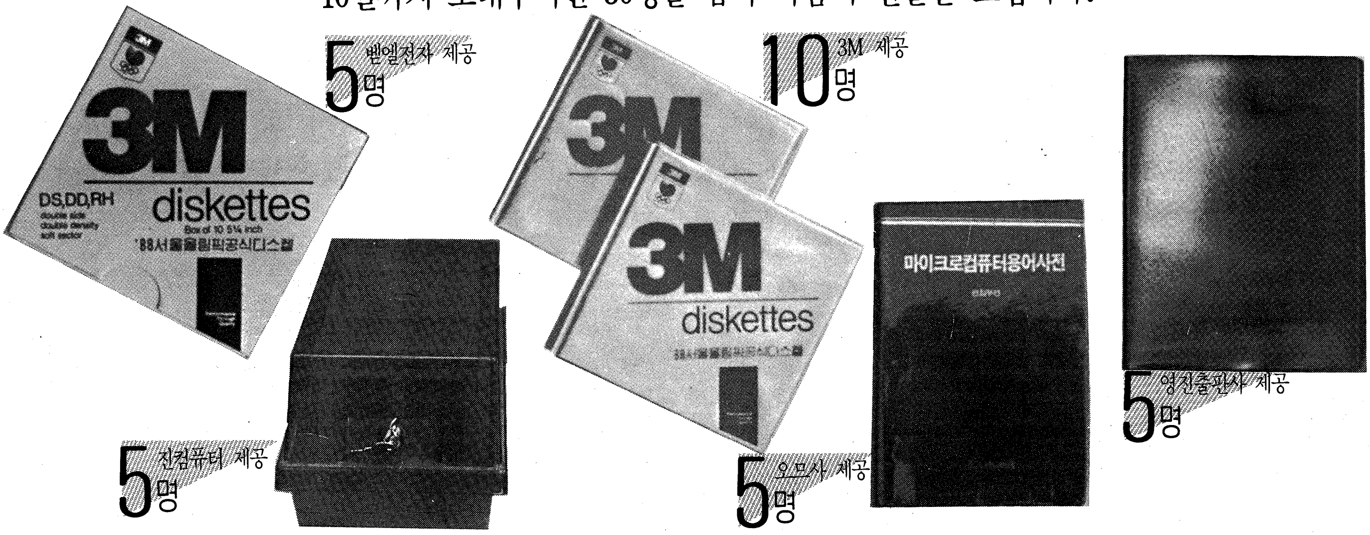
## 본체 386 Dragon 386UX



● 제조원 : 쌍용컴퓨터
● CPU : i80386
● 클럭 속도 : 25MHz
● 주기억용량 ROM : 64KB/RAM : 2MB
● 보조기억장치 FDD : 5.25″×1(1.2MB)
HDD : 5.25″×1(40MB)
● I/O 포트 수 직렬 : RS-232C(2개)
병렬 : 센트로닉스 타입(1개)
● 확장 슬롯 : 8개
● OS : MS-DOS, OS/2, UNIX V, 머지 386, XENIX
● 가격 : 660만원
● 기타 특성 : IBM PC XT/AT와 100% 확장성. EGA, CGA, MDA, 허큘레스 등의 다양한 화면 모드를 하나의 비디오 보드로 지원, 동일 보드상에서 한글/한자 처리.
● 문의처 : (02) 273-4416

# 애독자 선물대잔치

월간 퍼스널컴퓨터는 독자 여러분의 의견을 소중히 하여 책의 편집에 반영하고자 합니다.
관제엽서를 사용하여 아래의 질의서에 성의껏 답하시어
10일까지 보내주시면 30명을 뽑아 다음의 선물을 드립니다.




| | |
|---|---|
| ① 성명 | ② 나이 |
| ③ 주소 | ④ 전화번호 |
| ⑤ 보유기종 | ⑥ 사용용도 |
| ⑦ 사용언어 | ⑧ 사용경력 |
| ⑨ 책구입방법 | ⑩ 책구입동기 |
| ⑪ 이번호에 가장 좋았던 기사 | |
| ⑫ 이유 | |
| ⑬ 이번호에 가장 좋지 않았던 기사 | |
| ⑭ 이유 | |
| ⑮ 관심있는 분야 | ⑯ 게재를 원하는 기사 |
| ⑰ 컴퓨터에 관한 알고 싶은 질문 | |
| ⑱ 본지에 바라는 의견 | |

# 월간 퍼스널컴퓨터의 정기구독자에게는 아래와 같은 특전이 마련되어 있습니다.

—정기구독자는 본사가 보유한 각종 정보 및 자료를 무료로 배부받으실 수 있읍니다.
—본지에 게재되는 각종 프로그램을 디스켓에 담아 무료로 보내드립니다.
—본지 및 본지의 자매지(월간 전자과학) 주최로 개최되는 세미나 및 각종 행사에 무료로 참석하실 수 있읍니다.
—본지에서 행하는 각종 부대사업에 할인혜택이 주어집니다.
—구독기간중 책 가격이 인상되더라도 종전가격으로 구독이 가능합니다.

## ●구독신청은

—시중은행에서 지로용지를 이용하시거나 우체국에서 대체구좌 또는 소액환으로 송금하시면 됩니다.
은행 지로번호 : 3000667
우체국 대체구좌번호 : 010256—31—0526772

## ●정기구독료는

— 1 년 구독시 30,000원
— 6개월 구독시 15,000원

## ●구독신청시 주의사항은

-구독기간과 주소(우편번호), 전화번호와 받으실 분의 성명을 정확히 기입하시기 바랍니다.
—배달사고나 기타 변동사항은 본지 영업부로 즉시 연락하시기 바랍니다. 신고전의 분실사고는 책임지지 않습니다.

## ●정기구독 문의처는

—서울시 용산구 갈월동 17 ㊥ 140-150
월간 퍼스널컴퓨터 영업부
전 화 : (02)754—1044(대표)
FAX : 757—1049

## ■ 전국총판 ■

| | | |
|---|---|---|
| **●서울** | | |
| 종로, 중구 | 명문사 | 774—6314 |
| 성동, 동대문 | 홍익도서 | 446—3926 |
| 성북, 도봉 | 진원사 | 981—4884 |
| 강남, 강동 | 광일사 | 415—7330 |
| 관악, 동작 | 관악사 | 889—4504 |
| 서대문, 은평 | 형제사 | 304—9462 |
| 영등포, 구로 | 영도사 | 613—5381 |
| | 진명서점 | 272—5871 |
| | 관음서적 | 266—7792 |
| **●부산** | | |
| | 종합기술서적 | 462—2431 |
| | 영광도서전시관 | 805—9500 |
| **●대구** | | |
| | 한일서적 | 23—7101 |
| | 학원사 | 22—0336 |
| **●인천** | | |
| | 동양서점 | 73—3384 |
| **●광주** | | |
| | 삼복서점 | 22—0258 |
| **●경기** | | |
| 부천 | 대경서점 | 62—3672 |
| 안양, 안산 | 사랑글방 | 57—1916 |
| 수원 | 동아서점 | 5—6436 |
| **●강원** | | |
| 원주 | 원주종로서적 | 42—2943 |
| **●충북** | | |
| 충주 | 문학사 | 2—2814 |
| **●충남** | | |
| 천안 | 동방서적공사 | 63—3314 |
| 대전 | 동양서점 | 22—1538 |
| **●경북** | | |
| 포항 | 경북서점 | 2—2717 |
| 구미 | 중앙서점 | 52—7915 |
| **●경남** | | |
| 진주 | 동아서점 | 42—3768 |
| 진해 | 학애서점 | 2—2523 |
| 울산 | 우정서점 | 43—6003 |
| 마산 | 기술서적 | 56—1417 |
| **●전북** | | |
| 전주 | 홍지서점 | 6—5311 |
| 군산 | 동원서점 | 42—3776 |
| **●전남** | | |
| 목포 | 문화당 | 5—3086 |
| 여수 | 대양서점 | 62—2111 |
| **●제주** | | |
| | 광양서적 | 53—5077 |

# 책을 마감하고



한 20년쯤 뒤에는 **추억이** 되지 모른다는 생각에 기념으로—
(앞줄 왼쪽부터 이향선, 장현경, 주애경기자 뒷줄은 최영철, 강영도, 김용칠, 박생규, 박미현기자이다)

●지상에는 크고 작은 길들이 많이 있다.
그러나 도달점은 모두가 같다.
말을 타고 갈수도,
차를 타고 갈수도,
둘이서 셋이서도 갈 수가 있다.
그러나 마지막 한 걸음은
혼자서 걷게 마련이다.
그러므로, 아무리 어려운 일이라도
혼자서 하는 것 보다 더 나은 지혜나
능력은 없다.
라는 헤세의 "혼자"를 떠올리게 하는 진정 홀
로이고 싶고 생각하고프며 마무리 짓고자 하
는 시간입니다.
이 시점에서 새로이 잉태한 소중한 선물을
전해드립니다.
그리고, Merry christmas! P.C friend!
〈賢〉

●이제 무엇인가 마무리해야 할 것만 같은데
자꾸만 자꾸만 새롭게 또 달려가야 한다고
한다.
무엇인가 되기도 전에 부서져 버릴 것같은
쓰잘데 없는 생각만 맴돈다.
시작하면 거의 된 것이나 다름없다는 것은
영락없이 궤변일 수 밖에 없고 아직 '왠수덩이'
를 면하려면 까마득히 멀리 있을 뿐.
아직도 기다리고 있는 사람을 위해, 아직도
기다려야 할 사람을 위해 조용히 기도할 수 밖
에 없는 가난한 사람은 언제쯤 부자가 될 수
있을련지……〈仙〉

●하나!
백마디의 말은 절로 나오는데
한단어의 글은
왜 이리 쓰기 어렵노. 〈榮〉

●하늘이 일년 365일을 자기 몸이 갈라지는
듯한 고통을 감내함으로써 찬란한 빛을 발하는
태양을 떠오르게 하듯 우리는 이 한권의 책을
탄생시키기 위하여 많은 나날을 고통과 싸우며
견디어야 했다.
오직 그 하나를 위하여……〈七〉

●무지하게 빨리 달려온 느낌이다. 어느덧 코
트깃을 올려야할 만큼 차가운 바람이 분다.
사는일 자체가 미지 세계로의 도전의 연속이
라든가! 아뭏든 새로운 세계로의 행진은 무척
이나 힘이 든다.
걸음마를 처음 배우는 아가마냥. 한걸음 내
딛고 넘어지고, 다시 일어나 또 넘어지고. 마
주잡아 주려는 엄마의 따스한 손길도 마다한체,
다시 일어나 아장 아장, 한걸음 한걸음, 나
아가 씩씩하게 걸어나갈 수 있도록 많은 노력
을 해야 겠다. 〈鉉〉